

UNIVERSITÉ DE MONTRÉAL

INTÉGRATION D'UN SYSTÈME RADIO À BANDE ULTRA-LARGE POUR LA
NAVIGATION DE ROBOTS MOBILES

WASSIM RAFRAFI
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉLECTRIQUE)
JUIN 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

INTÉGRATION D'UN SYSTÈME RADIO À BANDE ULTRA-LARGE POUR LA
NAVIGATION DE ROBOTS MOBILES

présenté par : RAFRAFI Wassim

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. SAUSSIÉ David, Ph. D., président

M. LE NY Jérôme, Ph. D., membre et directeur de recherche

M. NERGUIZIAN Chahé, Ph. D., membre

REMERCIEMENTS

Je tiens à remercier tous ceux et celles qui ont contribué à la réalisation de ce projet, en particulier, Prof. Jérôme Le Ny pour avoir accepté d'assurer le bon déroulement de ce travail. Je me considère très chanceux d'avoir travaillé avec lui et je le remercie pour ses encouragements et ses conseils, dans la recherche et la vie universitaire qui m'ont été d'une aide précieuse.

Je désire remercier spécialement les professeurs Prof. Richard Gourdeau et Prof. David Saussié qui m'ont prodigué leurs conseils et m'ont accordé beaucoup de leur confiance.

Je souhaite aussi remercier Prof. Chahé Nerguizian pour avoir accepté d'être membre du jury d'évaluation de ce travail.

Je voudrais également exprimer mes remerciements aux personnels du département de génie électrique en particulier à M. Saad Chidami, M. Jacques Girardin, M. Bryan Tremblay et M. Marc Charboneau pour leur aide et leur soutien indispensables à l'élaboration du matériel nécessaire pour mes expériences.

Mes remerciements s'adressent aussi à Mahmoud Ghorbel qui a travaillé avec moi dans le cadre du cours Systèmes de Navigation du Prof. Jérôme Le Ny qui nous a été d'une très grande importance dans la réalisation de nos projets de maîtrise. À travers ce projet, Mahmoud a contribué au développement du système de localisation UWB/INS.

Je remercie également mes amis pour les bons moments qu'on a passés ensemble, spécialement Amine Tastouri, Corentin Germain, Reny Belguidoum, Ghazi Majdoub et Nouha Baroudi.

Enfin, je voudrais dédier ce travail à mes parents et mes deux chers frères qui n'ont jamais cessé de m'encourager. Je tiens également à exprimer ma reconnaissance à mon oncle et sa famille qui m'ont beaucoup soutenu.

RÉSUMÉ

La localisation revêt une grande importance dans les applications de robotique mobile. Cette tâche devient plus compliquée quand il s'agit d'une localisation d'un groupe de robots évoluant dans un environnement partagé. Un élément clé dans ce type d'application est le système de mesure des distances qui permet de localiser chaque robot en se basant sur les mesures des distances qui les séparent. Les travaux réalisés, dans la plupart des algorithmes multi-robots, supposent qu'il existe une solution efficace pour effectuer ces mesures de distance. Cependant, avoir une bonne précision dans le système de mesures représente un vrai problème, en particulier dans un environnement intérieur.

Le travail réalisé dans le cadre de cette maîtrise a pour objectif le déploiement d'un système permettant à un robot mobile d'effectuer des mesures de distance avec d'autres robots ou un certain nombre de points de référence. Ceci est fait en intégrant aux robots un système de communication Ultra Wideband (UWB) pour les mesures de distance. Cette technologie est l'une des plus intéressantes pour ce type d'application. Son avantage majeur est sa forte résolution temporelle pour effectuer des mesures assez précises du temps de propagation d'un signal.

Le système de mesure de distance développé a été également exploité dans la navigation robotique ce qui devrait être utile dans nombreuses applications. En effet, la fusion des mesures de distance UWB avec un système de navigation inertielle permet d'avoir un système de localisation haute précision qui peut être déployé autant à l'intérieur qu'à l'extérieur. Les résultats expérimentaux montrent que ce système de localisation hybride peut être intégré dans plusieurs types de plate-formes robotiques, tels que les drones et les robots au sol. En outre, le système conçu peut servir également à la communication courte portée entre les robots.

ABSTRACT

Localization is one of the most important tasks in mobile robotics applications. The complexity of this task increases when it comes to multi-robot localization. For such applications, a ranging system could be used to perform range measurements between robots and enable cooperative localization of the robots based on range measurements. Much of the work carried in multi-robot localization assumes that there is an efficient solution to perform ranging measurements. However, such a system is not easy to deploy, especially in an indoor environment.

In this report, we introduce a design of a ranging system based on Ultra-wideband technology. This system is able to provide effective tools for mobile robots to perform ranging measurements with other robots. UWB technology has important advantages for such applications due to its good ability to resolve the temporal features of signals.

The developed ranging measurement system is also used for robotic localization, which is useful in many applications, such as autonomous navigation. In fact, merging the UWB ranging system with an inertial navigation system allows us to achieve cm-level accuracy at high update rates. Merging these two systems makes up for the shortcomings of each, and gives a better performance than what is possible with a single system. The experimental results show that the hybrid positioning system can be used with several types of robotic platforms, such as drones and unmanned ground robots. In addition, the proposed system could be useful for the short-range communication between robots.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	v
TABLE DES MATIÈRES	vi
LISTE DES TABLEAUX	viii
LISTE DES FIGURES	ix
LISTE DES SIGLES ET ABRÉVIATIONS	xi
LISTE DES ANNEXES	xii
CHAPITRE 1 INTRODUCTION	1
1.1 Survol des solutions de localisation	2
1.2 Problème considéré	4
1.2.1 Objectifs de recherche	4
1.2.2 Localisation par ondes radio UWB	5
1.2.3 Fusion d'informations de localisation	9
1.3 Plan du mémoire	10
CHAPITRE 2 MESURE DES DISTANCES PAR TRANSMISSION UWB	11
2.1 Mesure de distance par mesure de temps de propagation du signal	11
2.1.1 Mesure de temps de propagation (<i>ToF</i>)	12
2.1.2 Algorithme de mesure de distance Double Two Way Ranging	15
2.2 Sources d'erreurs dans les mesures de distance	17
2.2.1 Comparaison entre l'algorithme TWR standard et l'algorithme double TWR	17
2.2.2 Retard du temps de propagation de signal	19
2.2.3 Perte de la ligne de vue entre les antennes	21
2.2.4 Impact de la directionnalité des antennes	23
2.3 Filtrage des mesures de distances	24
2.4 Conclusion	26

CHAPITRE 3	LOCALISATION UWB ET FUSION DES DONNÉES UWB/INS . . .	28
3.1	Localisation <i>UWB</i>	28
3.1.1	Méthode des moindres carrés	28
3.1.2	Filtrage de Kalman	30
3.1.3	Influence de la géométrie des balises sur la localisation	32
3.2	Localisation multi-capteurs	36
3.2.1	Système de navigation inertielle	36
3.2.2	Estimation de l'orientation avec l' <i>IMU</i>	39
3.2.3	Filtre de Kalman Étendu pour la fusion de données	41
3.2.4	Estimation de l'orientation par mesures <i>UWB</i> multiples	44
3.2.5	Performances de la localisation multi-capteurs	48
3.3	Conclusion	49
CHAPITRE 4	APPLICATIONS	51
4.1	Intégration du système de localisation <i>UWB</i> sur un nano quadricoptère . . .	51
4.1.1	Description de l'architecture logicielle et matérielle du quadricoptère .	52
4.1.2	Réalisation	52
4.2	Navigation 2D en l'absence de magnétomètre	54
4.2.1	Description de l'architecture du système	54
4.2.2	Réalisation	55
CHAPITRE 5	CONCLUSION	58
5.1	Synthèse des travaux	58
5.2	Limitations et futures perspectives de la solution proposée	59
RÉFÉRENCES	61
ANNEXES	64

LISTE DES TABLEAUX

Tableau 2.1	Temps de mesure de distance en fonction de la configuration du module UWB	16
Tableau 3.1	Taux d'erreur sur l'estimation de position par filtrage de Kalman avec les modèles P , PV et PVA	33
Tableau A1	Format général d'une trame	67
Tableau A2	Message de type <i>Poll</i>	67
Tableau A3	Message de type <i>Anchor Response</i>	68
Tableau A4	Message de type <i>Final Response</i>	68
Tableau A5	Message de type <i>Report</i>	68

LISTE DES FIGURES

Figure 1.1	Navigation d'un robot mobile entre deux points	1
Figure 1.2	Positionnement par différence entre les instants d'arrivée (TDOA) [Tiré de Liu et al. (2007a)]	6
Figure 1.3	Positionnement par angle d'arrivée (AOA)	7
Figure 1.4	Technique de localisation Fingerprinting.	8
Figure 1.5	Description du système multicauteurs	10
Figure 2.1	Réseau de nœuds UWB	12
Figure 2.2	Calcul de distance avec un échange de deux messages.	13
Figure 2.3	Calcul de distance avec un échange de trois messages.	14
Figure 2.4	Échange de message entre le <i>Tag</i> et l' <i>Anchor</i>	15
Figure 2.5	TWR [Tiré de DecaWave (2014a)]	18
Figure 2.6	double TWR [Tiré de DecaWave (2014a)]	19
Figure 2.7	Mesure d'une distance de 5m avec trois modulest	20
Figure 2.8	Retard du temps de propagation de signal [Tiré de DecaWave (2014b)]	21
Figure 2.9	Perte de la ligne de vue directe entre l'émetteur et le récepteur	22
Figure 2.10	Comparaison des mesures de distance	23
Figure 2.11	Caractérisation de l'erreur sur les mesures de distance	24
Figure 2.12	Résultats obtenus de l'algorithme de filtrage des mesures de distance	26
Figure 3.1	Localisation du Tag avec trois stations de base	29
Figure 3.2	Résultats de localisation obtenus avec l'algorithme de filtrage de Kalman sur une trajectoire carrée avec les trois modèles P , PV et PVA .	32
Figure 3.3	Influence de la géométrie des balises sur la précision de localisation .	33
Figure 3.4	Coefficient de dilution de précision en fonction de la géométrie des balises	35
Figure 3.5	Système de navigation inertielle (INS) [Borràs Sillero (2012)].	36
Figure 3.6	Repère de navigation du robot	38
Figure 3.7	Algorithme de descente de gradient [Madgwick et al. (2011)	40
Figure 3.8	Résultats d'estimation de l'orientation	41
Figure 3.9	Architecture du système de fusion de données [Borràs Sillero (2012)].	42
Figure 3.10	Résultats de simulation du système de fusion de données UWB/INS .	45
Figure 3.11	L'erreur sur l'estimation de l'orientation UWB en fonction de la longueur des côtés du triangle d	46
Figure 3.12	Résultats de la simulation du positionnement et de l'orientation avec les modules UWB.	48

Figure 3.13	Résultats de la simulation du positionnement avec les modules UWB.	49
Figure 3.14	Résultats de la simulation de l'orientation avec le système <i>UWB</i>	49
Figure 4.1	Intégration du module <i>DWM1000</i> sur le drone	51
Figure 4.2	Résultats des tests du système de positionnement UWB avec le Crazy- flie 2.0	54
Figure 4.3	Configuration matérielle du système de localisation multi-capteurs . .	55
Figure 4.4	Architecture des nœuds ROS	56
Figure 4.5	Résultats des tests du système de positionnement UWB avec le Husky	57
Figure A1	Description de la structure logicielle	65
Figure A2	Machine à états du <i>Tag</i>	66
Figure A3	Machine à états du Anchor.	66
Figure A4	Exemple d'une station du système de localisation	69

LISTE DES SIGLES ET ABRÉVIATIONS

UWB	Ultra Wide Band
IMU	Inertial Measurement Unit
ROS	Robot Operating System
MEMS	Micro-Electro-Mechanical Systems
GPS	Global Positioning System
ToF	Time of Flight
TWR	Two-Way Ranging
INS	Inertial Navigation System
SPI	Serial Peripheral Interface
EKF	Extended Kalman filter

LISTE DES ANNEXES

ANNEXE A	Architecture logicielle et matérielle du système UWB	64
----------	--	----

CHAPITRE 1 INTRODUCTION

La robotique mobile est un axe de recherche qui vise à automatiser certaines tâches dans l'intention de donner la capacité à un robot (terrestre ou aérien par exemple) de se mouvoir dans un environnement incertain et dynamique de manière autonome, c'est-à-dire sans intervention humaine. Pour effectuer une mission, un robot mobile doit pouvoir *naviguer* dans son environnement, c'est-à-dire se déplacer d'une position de départ, qui représente sa position actuelle, à un point d'arrivée spécifié par l'utilisateur. La navigation nécessite que le robot dispose d'un planificateur de trajectoire, voir Figure 1.1, qui lui-même doit connaître en tout temps la posture (position et orientation) du robot dans son environnement pour lui permettre d'atteindre sa cible. Cette posture du robot est fournie par un système de localisation, dont la précision doit être adaptée à la mission effectuée par le robot et à l'environnement dans lequel il évolue.

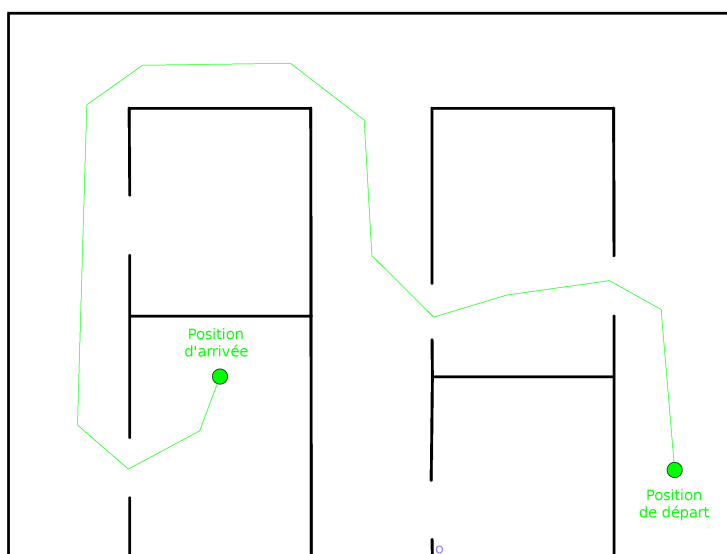


Figure 1.1 **Navigation d'un robot mobile entre deux points.** La trajectoire en vert décrit le chemin emprunté par le robot pour se déplacer entre la position de départ et la position d'arrivée.

Certaines technologies ont été largement étudiées dans la littérature et apportent des solutions éprouvées et efficaces au problème de la navigation dans des environnements extérieurs, comme le GPS. Pour les environnements intérieurs par contre (un bâtiment, un entrepôt...) on ne possède pas encore une technologie dominante. L'évolution massive des technologies

au cours de la dernière décennie a créé de nouvelles opportunités pour la localisation dans ce type d’environnement. Il serait ainsi intéressant de contribuer au domaine de recherche visant à développer un système de localisation qui répond au mieux aux particularités d’un environnement intérieur tout en gardant la possibilité de l’utilisation à l’extérieur.

Ce premier chapitre introductif présente dans un premier temps un certain nombre de solutions de localisation existantes, en mettant en avant les performances et les contraintes qui leur sont liées. Dans l’étape suivante, nous présentons le problème de localisation considéré dans le cadre de ces travaux.

1.1 Survol des solutions de localisation

Un grand nombre de travaux ont abordé la thématique de la localisation, compte tenu de l’importance qu’elle revêt dans une multitude de domaine, dont la robotique mobile. Ces travaux ont conduit à de nombreuses solutions de localisation au cours de la décennie passée. L’un des principaux facteurs de ce progrès est la réduction des coûts et de la taille des capteurs, en plus de la miniaturisation des ordinateurs qui facilite leur intégration dans les systèmes robotiques. En conséquence, il y a actuellement une forte tendance à élargir les milieux de l’évolution de ces systèmes automatisés notamment dans des secteurs comme l’automobile.

Le système de positionnement le plus répandu est le positionnement par satellite, ou GPS dans sa version américaine (Global Positioning System) [Groves (2013)]. Il est couramment utilisé dans des applications pour le positionnement extérieur, par exemple pour les automobiles. Néanmoins, ce système n’est pas opérationnel dans les environnements intérieurs, pour plusieurs raisons : d’une part, la précision du GPS standard est de l’ordre de 3 à 15 mètres, et d’autre part il y a une forte atténuation du signal à l’intérieur des bâtiments, ce qui rend ce système inapproprié pour un environnement intérieur dans la plupart des cas.

Certaines applications nécessitent une localisation plus fine, car localiser un robot à 10 mètres près dans un bâtiment est bien souvent inacceptable. Afin d’obtenir une meilleure précision, plusieurs technologies ont été explorées comme les systèmes de vision par ordinateur. Cette technologie est de plus en plus utilisée dans des applications robotique pour déterminer la position du robot et les objets caractérisant la zone d’exploration. Généralement, ces systèmes utilisent des caméras connectées à un ordinateur afin de prendre des images de l’objet à contrôler ou visualiser en temps réel. Ces caméras peuvent s’utiliser seules ou en paire pour former des systèmes de stéréo-vision [Tippetts et al. (2016)]. Grâce aux méthodes de segmentation des images, il est possible de repérer un objet dans le champ de vision de la

caméra. Ainsi la position de l'objet peut être calculée par des transformations entre l'image de la scène et les angles de vues de la caméra. Troiani et al. (2015) présente un système de localisation utilisant les données fournies par une caméra embarquée sur un quadricoptère, fusionnées avec une centrale inertielle (IMU). L'inconvénient de l'utilisation de la vision par ordinateur est la courte portée des caméras. D'autres problèmes se posent également comme l'identification des objets à localiser. Dans un scénario comme celui de la Figure 1.1, le système de localisation a besoin de repérer la position actuelle du robot ainsi que le point d'arrivée. Lorsque deux objets se croisent, l'un des objets masque l'autre pendant une courte période. Ainsi, le système de vision ne détecte qu'un seul objet dans la zone d'exploration ce qui peut affecter la tâche de contrôle du robot. De plus, au moment de la séparation des deux objets, il serait difficile de différencier entre les marqueurs qui représentent le robot et la cible.

Les technologies de communication à courte portée ont été explorées également pour la localisation dans des environnements intérieurs. Les auteurs de [Liu et al. (2007b)] présentent une vue d'ensemble des systèmes de localisation basés sur des technologies radio commerciales disponibles aux dates de la publication. L'omniprésence des équipements de radio, servant de relais de communication entre les objets mobiles, fait de ces technologies des candidates idéales pour le développement d'un tel système. De plus, leur prix étant relativement faible, ces technologies deviennent de plus en plus accessibles. Comme exemple, nous pouvons citer les réseaux locaux sans fil (WLAN) présents dans la plupart des environnements intérieurs et qui ont été largement utilisés à cette fin. L'une des premières mises en œuvre de ce type de système de localisation est basée sur les mesures de la puissance du signal reçu par l'antenne. Cette technique se repose principalement sur la mesure du rapport de puissance d'un signal par rapport à plusieurs points de référence pour pouvoir en déduire l'emplacement du robot. Généralement, cette méthode se déroule sur deux phases : une première phase qui consiste à construire une carte d'empreintes des puissances du signal dans plusieurs points de l'environnement. Ensuite, dans une seconde phase, l'estimation de position s'effectue en comparant les puissances des signaux reçus par rapport à la carte d'empreintes.

Ces systèmes utilisent aussi la méthode de triangulation pour l'estimation de la position du mobile par rapport à un certain nombre de balises radio qui servent comme point de référence. Cette méthode consiste à calculer la distance entre l'émetteur (celui qui veut se localiser) et les points de références. Plusieurs techniques ont été développées pour la mesure de ces distances [Groves (2013)], dont la technique de mesure du temps de propagation (time of flight) reste la plus simple et la plus utilisée. Ces algorithmes consistent à mesurer le temps de propagation d'un signal entre un émetteur et un récepteur. Les auteurs de [Vasishth et al. (2015)] présentent un système qui permet de localiser un robot équipé d'une carte WiFi avec

une précision de l'ordre de 1 m en utilisant la technique de mesure de temps de propagation. Ultra Wideband (UWB) est une technologie émergente dont les performances en termes de localisation semblent prometteuses, en raison de sa forte résolution temporelle par rapport aux systèmes à bande étroite et sa faible susceptibilité à l'évanouissement des signaux dans un environnement multi-trajet [Gezici et al. (2005); Bensky (2007); Wymeersch et al. (2012)]. Des travaux récents [Ledergerber et al. (2015); Mueller et al. (2015)] exposent des solutions de localisation basées sur cette technologie dont la précision obtenue est de l'ordre de quelques centimètres. Ledergerber et al. (2015) proposent un système qui permet à un robot mobile d'estimer sa position par rapport à un certain nombre de balises UWB synchronisées. Quant à Mueller et al. (2015), ils intègrent la localisation UWB sur un quadricoptère en utilisant des mesures de distance UWB fusionnées avec des mesures d'accélération et de vitesse angulaire, similaires à l'expérience présentée dans le chapitre 4. La compagnie Time Domain¹ propose aussi un système de localisation commercialisé basé sur la technologie UWB. Ils utilisent une technique appelée Tow Way Ranging pour l'estimation de la position du mobile. Elle consiste à effectuer plusieurs échanges de messages afin de calculer le temps de propagation d'un signal entre deux modules. La précision obtenue avec leur système est de l'ordre 2.1 cm quand les deux modules sont placées en ligne de vue directe et de l'ordre de 0.5 m sinon.

1.2 Problème considéré

Le problème de localisation en robotique mobile soulève plusieurs sous-problèmes. Il est donc important de définir le périmètre de ce mémoire et d'introduire les différentes technologies utilisées. Après la définition de nos objectifs de recherche, nous présentons une étude approfondie de la technologie UWB et les différentes techniques d'exploitation des signaux radio, afin de prévoir la meilleure solution de localisation à déployer pour remplir nos objectifs. Ensuite, nous présentons et commentons l'intégration d'une centrale inertielle IMU avec un transcepteur UWB.

1.2.1 Objectifs de recherche

Comme nous l'avons vu précédemment, les techniques de localisation existantes ne sont pas adaptées à tous les environnements. Ce projet s'intéresse particulièrement à la problématique de la localisation basée sur un système de mesure de distance. L'idée consiste à développer un système permettant à un robot mobile de se localiser dans son environnement en utilisant la technologie radio UWB. Cette technologie, prise individuellement, connaît des limitations

1. <http://www.timedomain.com/>

comme la perte de ligne de vue entre les différents équipements, ou le faible taux de rafraîchissement. Il est alors avantageux de fusionner au système UWB d'autres technologies de localisation. La fusion des données permet de diminuer les imperfections de chaque capteur en corrigeant la solution de localisation finale par celle provenant d'un autre capteur dont les performances sont meilleures dans certaines conditions. Une telle solution permettra de garantir une couverture du service de localisation plus importante avec une meilleure qualité sur l'estimation de la posture du robot.

La réalisation de cet objectif passe par deux autres sous-objectifs plus spécifiques :

- Concevoir un système de mesure de distance radio à base de modules UWB. Ceci passe par : le choix de la méthode d'extraction des informations des modules UWB et la réalisation des tests et la validation.
- Proposer et implémenter un algorithme qui permet de fusionner les données du système de localisation UWB avec le système de navigation inertielle INS.

1.2.2 Localisation par ondes radio UWB

UWB est l'une des technologies de communication sans fil les plus compétitives pour la localisation intérieure, notamment pour ses caractéristiques très intéressantes en ce qui concerne la complexité de l'utilisation, le coût de la réalisation, la couverture et la précision. Cette technologie permet la transmission de données numériques sur une largeur de bande de 500 MHz ou plus avec une faible puissance. À la différence des systèmes RFID classiques, qui fonctionnent sur des bandes étroites, la technologie UWB émet simultanément des signaux sur plusieurs bandes de fréquences qui s'étalent de 3.1 à 10.6 GHz. Une façon d'obtenir un tel contenu spectral large est de générer des impulsions de très courte durée, inférieure à la nanoseconde. Cette forte résolution temporelle permet d'effectuer des mesures de temps de propagation d'un signal relativement précise entre un émetteur et un récepteur. Elle peut également être utilisée à proximité d'autres signaux RF sans provoquer d'interférences en raison des différences dans les types de signaux et du spectre utilisés. De plus, le système UWB est moins sensible aux phénomènes de réflexion, réfraction ou diffraction d'onde dans un environnement multi-trajet, contrairement aux systèmes à bande étroite, ce qui réduit les erreurs lors des calculs des temps de propagation.

Les caractéristiques techniques du système UWB semblent bien adaptées pour des applications de localisation. Le positionnement d'un nœud mobile avec cette technologie repose principalement sur les informations provenant des signaux radio circulant entre l'émetteur (celui qui veut se localiser) et un certain nombre de points de référence. Les techniques de localisation associées à cette technologie sont généralement basées sur les techniques tem-

porelles, par exemple la méthode TOA (Time Of Arrival) et TDOA (Time Difference Of Arrival) [Caffery Jr (2006)]. Ces méthodes peuvent être divisées en trois catégories :

— **Mesure de temps de propagation :**

Les systèmes de localisation utilisant ces techniques reposent sur le calcul du temps de propagation des signaux radio circulant entre l'émetteur et les stations de base. Si tous les nœuds UWB sont synchronisés dans le temps, alors l'émetteur peut déterminer les distances qui le séparent des points de référence par la méthode du temps d'arrivée (Time Of Arrival ou TOA) [Shimizu and Sanada (2004)], ainsi que sa position par un algorithme de trilatération semblable à celui utilisé pour le GPS. Une autre technique couramment utilisée est la différence de temps d'arriver (Time Difference Of Arrival ou TDOA) qui consiste à mesurer les différences entre les temps d'arrivée des signaux au niveau des récepteurs [Caffery Jr (2006)]. La balise UWB placée sur le mobile émet périodiquement des impulsions, qui sont reçues par des récepteurs situés dans des positions connues dans l'environnement, voir Figure 1.2. Étant donné que les stations de base sont parfaitement synchronisées entre elles, la différence dans le temps d'arrivée de l'impulsion de chaque balise donne l'emplacement du robot.

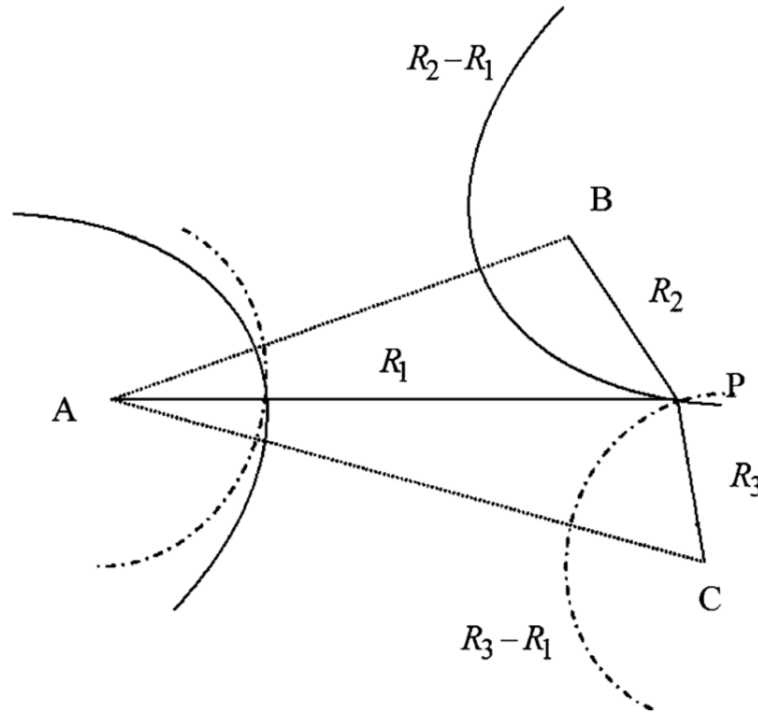


Figure 1.2 Positionnement par différence entre les instants d'arrivée (TDOA) [Tiré de Liu et al. (2007a)]

Cependant, dans la majorité des cas la tâche de synchronisation n'est pas aisée. Une

alternative est d'utiliser la méthode Two Way Ranging (TWR) qui ne nécessite pas de synchronisation entre l'émetteur et les récepteurs. Cette technique consiste à effectuer plusieurs échanges de messages afin de calculer le temps d'aller-retour et en déduire ensuite le temps de propagation du signal (Time of Flight ou ToF).

— **Angle d'arrivée (Angle Of Arrival ou AOA) :**

La technique AOA consiste à mesurer l'angle d'arrivée d'une onde provenant du mobile par rapport à une direction prédéfinie [Niculescu and Nath (2003)], comme le montre la Figure 1.3.

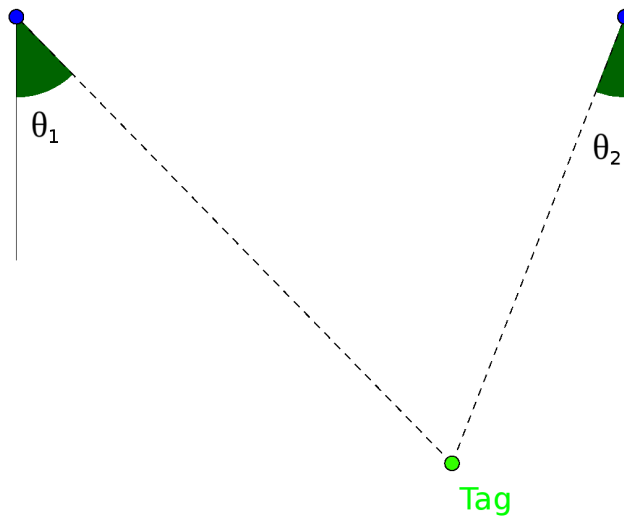


Figure 1.3 Positionnement par angle d'arrivée (AOA)

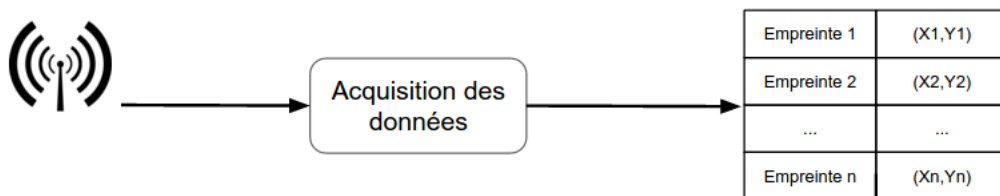
Pour déterminer l'emplacement du mobile dans une modélisation en deux dimensions du problème, il suffit de mesurer l'angle d'arrivée par rapport à deux points de références placée à une distance de séparation connue. Les signaux transmis par le mobile sont détectés en utilisant des antennes directionnelles. Cette technique est inadaptée aux environnements dans lesquels de nombreux multi-trajets sont présents car un signal réfléchi ne représente pas l'angle à partir duquel il a été transmis. La précision de cette technique dépend aussi de la précision de l'antenne pour détecter l'angle d'arrivée de l'onde. Une faible variation de l'angle peut engendrer une erreur non négligeable sur l'estimation de l'emplacement réel du mobile.

— **Fingerprint :**

Cette approche de localisation repose sur une étude du champ d'opération du système de localisation. Cette étude consiste à recueillir des informations sur les caractéris-

tiques des signaux reçus à des positions différentes. Pour constituer une empreinte du signal à une position connue, plusieurs types d'informations peuvent être utilisés, tels que la mesure du rapport de puissance d'un signal et le profil moyen de puissance de la réponse impulsionnelle. Cette technique se déroule sur deux phases, voir Figure 1.4 : une phase d'apprentissage ou hors-ligne et une phase de localisation.

Phase d'apprentissage



Phase de localisation

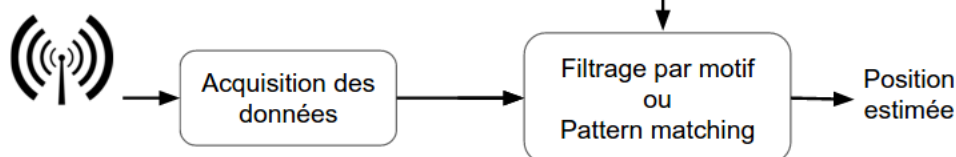


Figure 1.4 Technique de localisation Fingerprinting.

La phase d'apprentissage consiste à recueillir des empreintes du signal à chaque emplacement dans la zone d'exploration du robot et de construire ensuite une carte d'empreinte caractérisant le signal dans plusieurs points de l'environnement. Cette carte d'empreintes est une base de données des points prédéfinis couplés avec diverses caractéristiques du signal radio, par exemple la force du signal ou le temps de propagation entre les deux équipements. Dans la phase de la localisation, l'emplacement du robot est déterminé en comparant les caractéristiques du signal radio à point inconnu par rapport à la carte d'empreintes. Plusieurs algorithmes ont été développés pour faire la correspondance entre les caractéristiques du signal mesuré et la carte d'empreinte sauvegardée dans la phase d'apprentissage tel que la méthode des k plus proches voisins et les réseaux de neurones artificiels.

L'utilisation de cette méthode ne nécessite pas de matériel spécifique ni de synchronisation dans le temps entre les équipements radio. Le système peut être une application logicielle ce qui permet de réduire la complexité et les coûts de manière significative par rapport aux techniques de localisation temporelles ou d'angle d'arrivée. Cette

technique est principalement utilisée dans des environnements multitrajectoires où les signaux peuvent être soumis à des réflexions, réfractions ou diffractions avant d’atteindre l’antenne. Par exemple dans les mines, les signaux sont réfléchis plusieurs fois avant d’atteindre le module radio, ce qui peut causer un affaiblissement du signal. Les auteurs de [Nerguizian et al. (2006)] propose un système de localisation en utilisant la technologie UWB utilisé dans une mine qui repose sur la technique de Fingerprint. Les expériences ont montré que le système peut atteindre une précision de localisation de l’ordre de 2 m dans un tel environnement.

L’inconvénient majeur de cette approche est le processus d’étalonnage laborieux. L’ajout des nouvelles stations nécessite de refaire la phase d’apprentissage. En outre, un simple changement de l’environnement, tel que le déplacement d’un objet se traduit par des changements dans la propagation du signal qui peuvent affecter la précision de la localisation. Afin de maintenir la précision de positionnement, le processus de calibration doit être répété périodiquement.

1.2.3 Fusion d’informations de localisation

Un système de localisation UWB se base sur des mesures de distance provenant d’un échange de plusieurs messages avec chaque balise. Pour avoir une estimation de position du mobile dans un plan en deux dimensions, le système de localisation aura besoin d’effectuer des mesures de distance avec au moins trois balises. De ce fait, cette technologie a un relativement faible taux d’échantillonnage, qui peut limiter la vitesse de déplacement du mobile. Un autre inconvénient est que le système UWB avec un seul nœud mobile n’est pas en mesure de fournir des informations d’attitude. Plusieurs émetteurs sont nécessaires sur le mobile pour une mesure d’orientation, ce qui réduit encore plus le taux d’échantillonnage. Par contre, les erreurs obtenues avec ce système sont bornées, il n’y a pas de divergence au cours du temps.

À l’inverse, un système de navigation inertielle est capable de fournir un taux de rafraîchissement beaucoup plus élevé, et les mesures de position fournies par ce système ont l’avantage de ne pas dépendre d’autres sources extérieures. Toutefois, à cause des problèmes de dérive il y a une dégradation de la précision au fil du temps. La fusion de ces deux systèmes compense les lacunes de chacun, et donne une meilleure performance que celle fournie par chaque système pris individuellement. La question qui se pose alors est de trouver la méthode appropriée pour la fusion des données provenant de deux systèmes différents et de modéliser le système afin d’obtenir une meilleure estimation de position qui suit la dynamique de déplacement du mobile, voir la Figure 1.5.

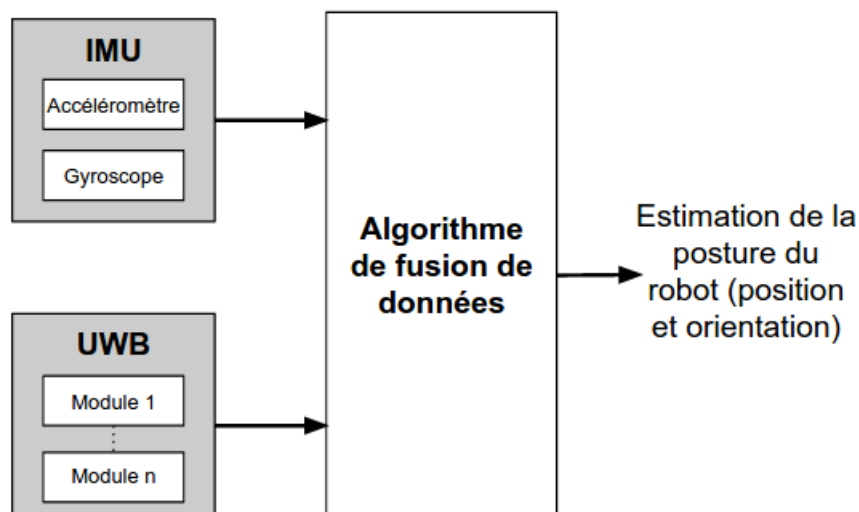


Figure 1.5 Description du système multicapteurs

1.3 Plan du mémoire

Au **chapitre 2**, nous présentons la première contribution de ce projet : le système de mesure des distances UWB. Nous commençons par détailler les algorithmes de mesure de distances par mesure de temps de propagation d’une impulsion UWB. Nous présentons ensuite les sources d’erreur qui peuvent affecter les mesures de distance UWB ainsi qu’un algorithme de filtrage de données.

Au **chapitre 3**, nous décrivons d’abord les techniques (classiques) de localisation par “ranging” utilisées pour le développement de notre système : méthode de Gauss-Newton et filtre de Kalman. Puis, nous expliquons en détail le fonctionnement de la localisation multi-capteur et l’estimation de l’orientation avec des modules UWB.

Au **chapitre 4**, nous validons finalement le système de positionnement UWB et les algorithmes développés en présentant les résultats obtenus dans deux cas d’applications de ces notions sur deux types de robots. Nous décrivons dans un premier temps l’intégration du système sur un quadricoptère et nous présentons ensuite l’application du système de localisation sur un robot au sol ainsi que l’estimation de l’orientation de ce robot.

CHAPITRE 2 MESURE DES DISTANCES PAR TRANSMISSION UWB

Dans ce chapitre, nous décrivons le système de ranging radio-fréquence utilisé, basé sur des transcepteurs UWB de la société Irlandaise DecaWave, appelés ScenSor *DW1000*. D’abord, nous expliquons en détail le fonctionnement des modules UWB utilisés pour effectuer les mesures de distance. Une fois les balises installées, le système de localisation UWB peut être utilisé tel quel pour obtenir une mesure de position, de la même façon qu’un système GPS. Nous présentons ensuite les sources d’erreurs qui peuvent affecter les mesures de distance UWB. Nous concluons en présentant un algorithme de filtrage des mesures de distance afin d’obtenir une meilleure précision. L’architecture matérielle système et l’implémentation de l’algorithme développé sont détaillées dans l’ANNEXE A.

2.1 Mesure de distance par mesure de temps de propagation du signal

Une composante principale du système développé est un système de localisation par mesure de distances (ranging) par rapport à un ensemble de balises radio fixes placées à des positions connues. Ces distances sont déduites de mesures du temps de propagation d’un signal radio entre les balises et le mobile. Afin d’obtenir une précision suffisante, la mesure de ce temps de propagation doit être faite avec une précision meilleure que 1 ns (une erreur de 1 ns correspond à une erreur sur la distance de 30 cm). Les systèmes de communication UWB sont bien adaptés pour faire des mesures avec ce niveau de résolution temporelle [Gezici et al. (2005); Bensky (2007)]. Ces mesures de distance peuvent être également exploitées par plusieurs types d’applications dont la plus simple est de déterminer la distance entre deux points afin d’établir un réseau composé par plusieurs robots, comme le montre la Figure 2.1. Une telle situation peut se présenter dans le cas des catastrophes naturelles où les services d’urgence sont les premiers intervenant sur les lieux. Au lieu de risquer la vie des sauveteurs, un tel système nous permet d’envoyer un groupe de robots à l’intérieur du bâtiment pour évaluer son état, de détecter la présence de survivants, de gaz potentiel ou de fuites de produits chimiques. Cependant, dans la majorité des cas il n’y a aucune infrastructure de localisation temps réel disponible, donc les robots doivent se localiser de manière coopérative. Cette méthode consiste à estimer les positions de tous les robots dans le réseau en combinant les mesures de distance entre des paires de robots, en supposant que suffisamment de mesures sont disponibles.

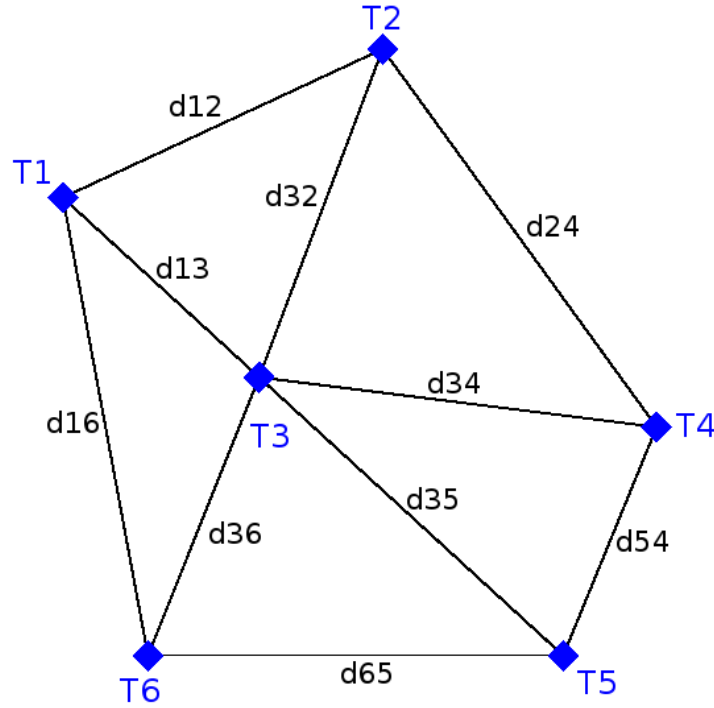


Figure 2.1 Réseau de nœuds UWB

2.1.1 Mesure de temps de propagation (*ToF*)

Le principe de fonctionnement de ce type de système de mesure de distance entre un émetteur et un récepteur est basé sur un échange de messages entre le *Tag* et les différentes *Anchors* en utilisant une variété de techniques, dans le but de calculer la distance qui sépare le *Tag* des différentes *Anchors* et d'en déduire ainsi son emplacement. De manière générale il y a deux façons possibles de mettre en œuvre un tel système.

Synchronisation d'horloges des émetteurs et récepteurs

Dans une configuration où l'émetteur et les récepteurs sont synchronisés dans le temps, le *Tag* diffuse un message à toutes les stations en même temps à un instant connu. À la réception du message au niveau des *Anchors*, ces derniers calculent le temps de propagation du signal ainsi que la distance qui les sépare de l'émetteur.

La précision des mesures de distance utilisant ce système dépend crucialement de la synchronisation entre les différents nœuds radio utilisés (émetteurs et récepteurs), en raison du fait qu'un simple défaut de synchronisation entre les différents nœuds du système peut causer

une erreur importante sur l'estimation de position. De plus, l'étape de synchronisation n'est pas toujours aisée et peut engendrer des coûts de réalisation élevés, ce qui rend ce système peu utilisé.

Émetteur et récepteur non synchronisé

Dans le cas d'un système où les horloges de différents noeuds ne sont pas synchronisées, le *Tag* fait un échange de plusieurs messages avec chaque *Anchor*. Cette méthode nécessite que les nœuds radio soient capables de recevoir ainsi que de transmettre des messages, ce qui implique une consommation d'énergie plus importante par rapport à la méthode de synchronisation d'horloges des émetteurs et récepteurs. De manière générale, pour avoir une mesure de distance entre deux modules radio non synchronisés, un émetteur et un récepteur, nous n'avons besoin que de deux échanges de messages comme le montre la Figure 2.2.

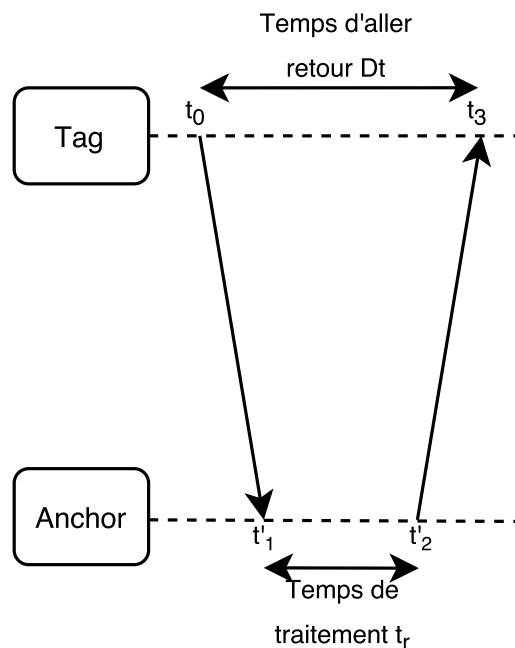


Figure 2.2 Calcul de distance avec un échange de deux messages.

1. Le *Tag* envoie un message à l'*Anchor* et note le temps d'envoi t_0 .
2. Lorsque l'*Anchor* reçoit un message, il note le temps réception t'_1 .
3. Ensuite l'*Anchor* envoie la réponse et note le temps d'envoi t'_2 .
4. À la réception de la réponse de l'*Anchor*, le *Tag* note le temps de réception t_3 et lance le calcul de la distance.

À la fin de l'échange, le *Tag* peut calculer le temps d'aller-retour $Dt = t_3 - t_0$ (le temps qu'il a fallu pour compléter l'échange des messages) qui inclut le délai de traitement entre la réception du message du côté de l'*Anchor* et la transmission de la réponse $t_r = t'_2 - t'_1$. Ainsi, ToF est obtenu par la formule suivante :

$$ToF = \frac{Dt - t_r}{2}$$

Enfin, la distance peut être calculée en multipliant le temps d'un aller simple par la vitesse de la lumière. Cependant, pour avoir une meilleure précision de mesure de distance il faut minimiser la dérive entre les horloges des deux modules.

La Figure 2.3 présente une solution qui peut remédier à ce problème, où l'*Anchor* envoie deux réponses à la suite d'une réception d'un message du *Tag*. Il envoie une première réponse, puis il se met en attente pour la même durée t_r et envoie son deuxième message. Ainsi, le *Tag* peut calculer le temps de traitement t_r avec sa propre horloge et en déduit ensuite le temps d'aller simple :

$$ToF = \frac{Dt - 2 * t_r}{2}$$

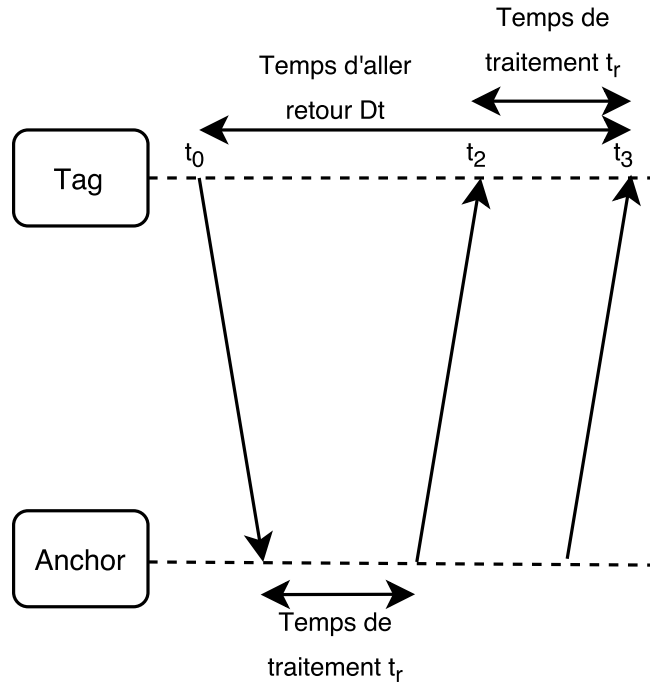


Figure 2.3 Calcul de distance avec un échange de trois messages.

2.1.2 Algorithme de mesure de distance Double Two Way Ranging

L'algorithme développé dans le cadre de ce projet pour mesurer la distance entre deux équipements UWB est basé sur un échange de messages suivant un protocole de TWR modifié, qui ne nécessite pas de synchroniser les différents modules. Tout d'abord, l'*Anchor* se met à l'écoute des requêtes provenant du *Tag*. Ensuite, le *Tag* lance un échange en envoyant une séquence qui identifie l'*Anchor* via un système d'adressage. Le récepteur envoie alors une réponse à l'émetteur. Une fois que le *Tag* a reçu la réponse, il est capable de mesurer le temps d'aller-retour.

La Figure 2.4 présente les échanges de messages entre le *Tag* et l'*Anchor*, en cinq étapes :

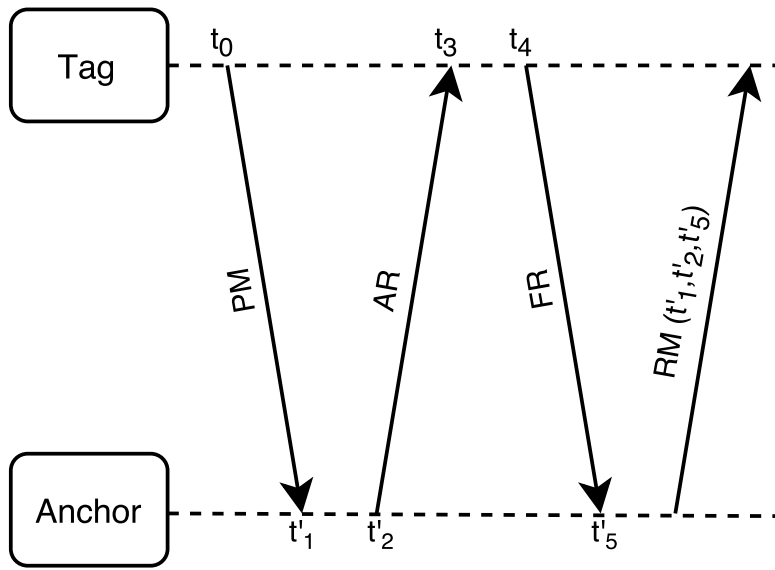


Figure 2.4 Échange de message entre le *Tag* et l'*Anchor*.

1. Le *Tag* envoie un message de type *Poll* à l'*Anchor* et note le temps d'envoi t_0 . Il se met ensuite à l'écoute d'un message *Anchor Response* de la part de l'*Anchor*. Si aucune réponse n'arrive au *Tag* avant l'écoulement du temps alloué, l'échange courant sera interrompu et il envoie un nouveau message *Poll*.
2. Lorsque l'*Anchor* reçoit un message de type *Poll* qui lui est adressé, il note le temps de réception t'_1 et envoie un message de type *Anchor Response* au *Tag* et note le temps d'envoi t'_2 .
3. Lorsque le *Tag* reçoit la réponse de la part de l'*Anchor*, il note à son tour le temps d'arrivée du message t_3 , envoie un message de type *Final Response* et note le temps d'envoi t_4 .

4. Quand l'*Anchor* reçoit le message final de la part du *Tag*, il note le temps de réception t'_5 . À ce stade, nous avons tout le nécessaire pour calculer le temps d'aller simple (*ToF*) après avoir sauvegardé tous les timestamps de l'échange. L'*Anchor* envoie donc son dernier message du type *Report* en ajoutant dans le champ de données du paquet les timestamps t'_1 , t'_2 et t'_5 afin de permettre au *Tag* de calculer la distance.
5. Une fois que le *Tag* a reçu le dernier message de l'*Anchor* qui contient les timestamps, il calcule le temps *ToF* selon la formule suivante

$$ToF = \frac{(t_3 - t_0) - (t'_2 - t'_1) + (t'_5 - t'_2) - (t_4 - t_3)}{4}$$

$$Distance = ToF * speed\ of\ light.$$

Notons en passant que lorsqu'un module envoie un message à un instant t_0 , il peut enregistrer ce timestamp mais pas envoyer cette valeur dans son message, car t_0 correspond à l'instant où le message quitte l'antenne du module.

En utilisant ce protocole, nous avons pu obtenir expérimentalement un temps de 3 ms pour faire une mesure de distance entre deux modules. En effet, selon le débit de communication et la taille de champ de données, le temps de propagation d'un message entre deux modules *DWM1000* peut varier de 110 μ s à 4 ms. Cela signifie que le temps de calcul d'une distance peut varier de quelques millisecondes à quelques dizaines de millisecondes. Le tableau 2.1 illustre un exemple des temps de mesure de distance obtenus pour deux configurations possibles des modules *DWM1000*.

Tableau 2.1 Temps de mesure de distance en fonction de la configuration du module UWB

Temps de mesure de distance	Débit de communication	Taille des paquets
3 ms	6,81 Mbps	128 octets
17 ms	110 kbps	1024 octets

2.2 Sources d'erreurs dans les mesures de distance

Les mesures de distance UWB sont généralement affectées par un bruit de 10 cm, en plus d'une erreur par rapport à la distance réelle d'une dizaine de centimètres. Les erreurs qui affectent les mesures de distance proviennent de plusieurs sources comme par exemple la dérive des horloges des modules UWB et le montage physique de la puce *DWM1000* et de l'antenne. Afin d'avoir une solution robuste, ces erreurs doivent soit être éliminées ou contrôlées en ajustant certains paramètres qui peuvent causer des erreurs non négligeables sur les mesures de distance.

Cette section a pour but d'effectuer un panorama des sources d'erreurs affectant la précision des mesures de distance. Nous commençons par une comparaison entre l'algorithme de *Two-way ranging* utilisé et la méthode standard en mettant en avant les performances et les contraintes qui leur sont liées. Nous présentons ensuite le problème du retard du temps de propagation du signal, ainsi qu'un algorithme d'étalonnage des modules. Nous finissons par le problème de perte de ligne de vue entre les modules *DWM1000*.

2.2.1 Comparaison entre l'algorithme TWR standard et l'algorithme double TWR

Une des sources d'erreur qui affecte considérablement les mesures de distance est la dérive des horloges des modules UWB. Du fait que les horloges des émetteurs et des récepteurs ne sont pas synchronisées dans le temps, il est difficile de calculer le temps de traitement au niveau des récepteurs tout en respectant l'horloge de l'émetteur. DecaWave (2014a) présente une étude sur l'impact de la dérive des horloges en utilisant la méthode présentée dans la section 2.1 et l'algorithme que nous avons implémenté, détaillé dans la section 2.1.2.

Considérons le schéma de la Section 2.1, comme le montre la Figure 2.5, le *Tag* active l'échange en envoyant un message au récepteur. Ce dernier envoie une réponse après un certain temps de traitement. Ainsi, l'erreur sur le temps de propagation de signal en fonction de la dérive des horloges peut être calculée avec la formule suivante :

$$erreur = \frac{1}{2} t_{replyB} (e_A - e_B)$$

Où t_{replyB} est la différence de temps entre la réception d'un message et l'envoi de la réponse au niveau du récepteur. e_A et e_B représentent l'erreur de fréquence des horloges respectivement à l'émetteur et au récepteur.

Nous pouvons donc remarquer qu'il y a une forte dépendance sur le temps de traitement

t_{replyB} dans cette équation. La Figure 2.5b présente le niveau d'erreur obtenu en fonction de la différence de la dérive des horloges de l'émetteur et du récepteur. Nous remarquons qu'une variation du temps de traitement de $10\ \mu s$ à $1000\ \mu s$ au niveau du récepteur peut engendrer une erreur de l'ordre de quelques dizaines de centimètres à une centaine de centimètres.

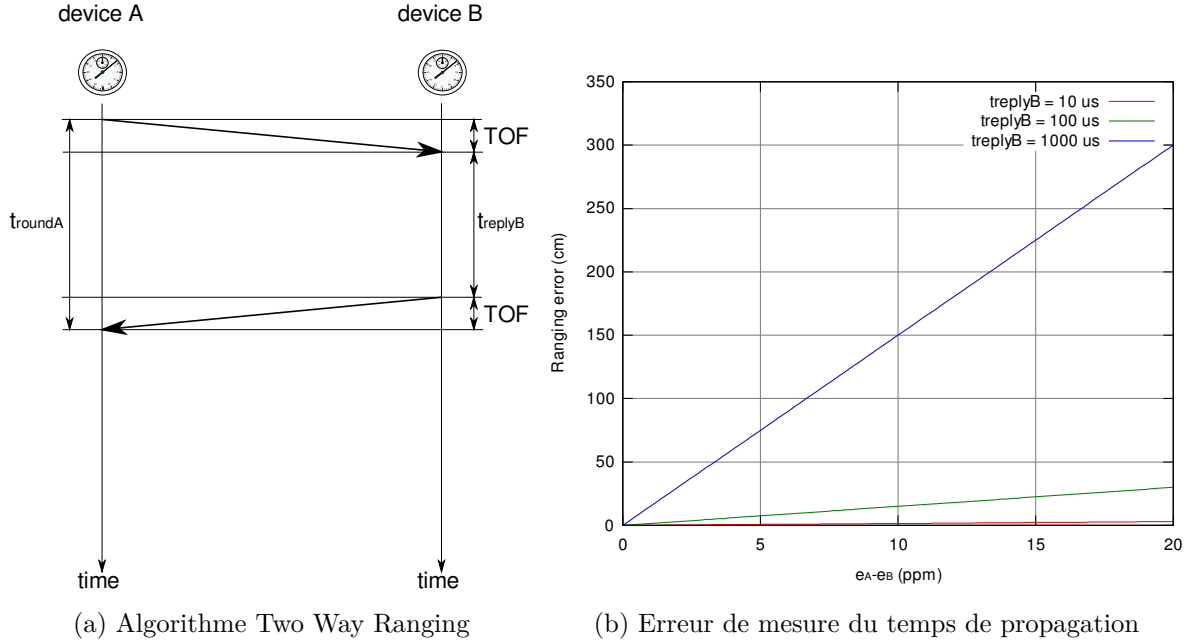


Figure 2.5 TWR [Tiré de DecaWave (2014a)]

Le schéma de la Section 2.1.2 présente une solution pour s'affranchir du problème de la dérive des horloges et réduire l'impact du temps de traitement entre la réception et la transmission d'un message afin d'obtenir de meilleurs résultats. En effet, l'ajout d'un message sur l'algorithme TWR standard permet de calculer le temps de propagation du signal entre les deux équipements des deux côtés, émetteur et récepteurs. Ainsi, l'erreur sur la mesure de distance s'exprime comme suit [DecaWave (2014a)] :

$$erreur = \frac{1}{2} \Delta_{reply} (e_A - e_B)$$

Avec ce schéma, l'erreur dépend de Δ_{reply} qui représente la différence entre le temps de traitement du côté de l'émetteur t_{replyA} et du côté du récepteur t_{replyB} . La Figure 2.6b nous montre que l'erreur ne varie que d'une dizaine de centimètres en fonction de la dérive des horloges des modules *DWM1000*.

Le principal inconvénient de l'utilisation de cette méthode est la consommation de puissance qui est plus élevée par rapport à la méthode standard. De plus, avec la difficulté de reporter le

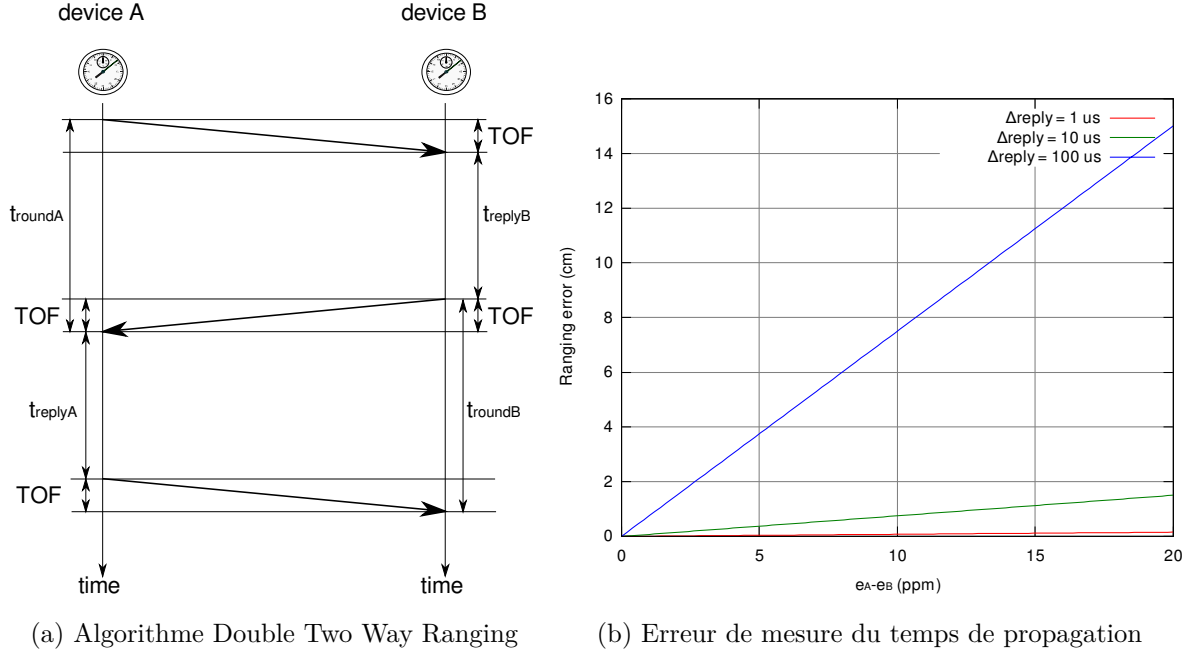


Figure 2.6 double TWR [Tiré de DecaWave (2014a)]

temps d'envoi d'un message des stations de base vers l'émetteur, l'ajout d'un quatrième message s'avère nécessaire. Ceci réduit la fréquence atteignable pour la mise à jour des mesures de distance.

2.2.2 Retard du temps de propagation de signal

Une autre source d'erreur qui affecte les mesures de distances est le retard de propagation interne entre le module *DW1000* et l'antenne. La Figure 2.7 présente des mesures de distances brutes que nous avons obtenues expérimentalement avec différents modules *UWB* placés à une distance réelle égale à 5 m.

L'erreur de mesure de distance peut varier d'un module à un autre à cause du montage physique de l'antenne et de la puce *DW1000* qui n'est pas forcément le même sur chaque module. Ainsi, ces variations peuvent introduire des erreurs dans les mesures de distance d'une dizaine de centimètres. Comme nous l'avons vu précédemment, l'algorithme de mesure de distance se base sur le calcul de temps propagation de signal entre deux modules en utilisant les timestamps sauvegardés à l'envoi où à la réception de chaque message. Or comme le montre la Figure 2.8, ces timestamps comprennent le retard de propagation entre la puce *DW1000* et l'antenne. En conséquence, le temps de propagation du signal mesurée $ToF_{mesurée}$

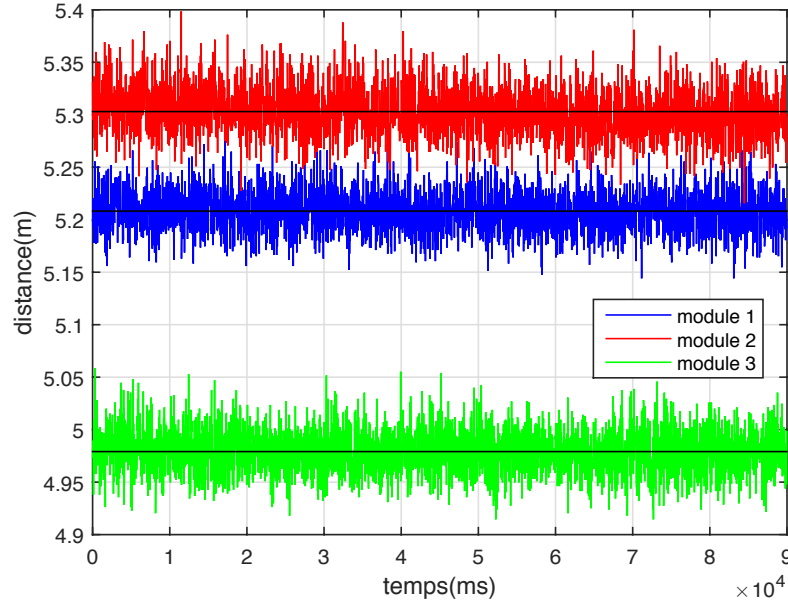


Figure 2.7 **Mesure d'une distance de 5m avec trois modules.** On voit ici que les modules UWB sont associés initialement à différents niveaux de biais (erreur systématique) dans la distance mesurée à cause du retard du temps de propagation de signal interne. Les lignes noires représentent la valeur moyenne du signal reçu par les différents modules. Nous pouvons remarquer aussi cette erreur systématique peut varier d'un module à un autre.

s'exprime comme suit :

$$ToF_{mesurée} = t_{RX} + ToF + t_{TX}$$

Avec t_{TX} représente le retard de transmission du message et t_{RX} représente le retard de la réception du message.

Ainsi, une calibration s'avère nécessaire afin d'obtenir la précision escomptée dans l'application finale. DecaWave (2014b) présente un algorithme de calibration des modules UWB. Cette méthode d'étalonnage peut donner une précision de 4.5 cm environ entre les différents modules. Une configuration minimale pour l'exécution de l'algorithme consiste à mesurer les distances entre au moins trois modules. Ces mesures sont ensuite sauvegardées dans la matrice *EDM* qui s'exprime comme suit :

$$EDM = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix}$$

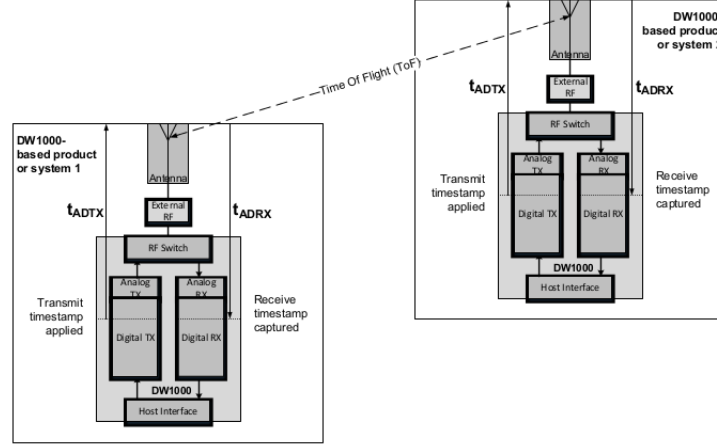


Figure 2.8 Retard du temps de propagation de signal [Tiré de DecaWave (2014b)]

Où d_{ij} est la distance entre la puce i et la puce j et n le nombre de puces à calibrer.

Le retard de propagation peut être estimé en minimisant la différence entre la matrice des distances réelles et la matrice des distances mesurées :

$$\min \| EDM_{\text{réelle}} - EDM_{\text{mesurée}} \|.$$

Ici $EDM_{\text{réelle}}$ est donnée par les distances réelles entre les différents modules et $EDM_{\text{mesurée}}$ est la matrice des mesures de distance UWB qui séparent les modules $DWM1000$.

Notons aussi que la température peut faire varier le retard de propagation du signal entre la puce et l'antenne de $2.15 \text{ mm}/^{\circ}\text{C}$. Nous recommandons donc que la température à laquelle la calibration a été effectuée soit enregistrée afin d'ajuster les paramètres des mesures de distances.

2.2.3 Perte de la ligne de vue entre les antennes

La perte de ligne de vue entre les modules radio (Non-Line-Of-Sight ou NLOS) est un problème très connu dans la littérature. UWB, comme toutes les autres technologies radio, souffre d'une atténuation de la force de signal dans un environnement multi-trajets. Dans un milieu intérieur, la présence de plusieurs objets dans la zone d'exploration (les murs, les meubles, etc) peut apporter des réflexions, des réfractions ou des diffractions sur les impulsions UWB. Ces effets provoquent des trajets multiples qui interfèrent entre eux et perturbent le niveau de signal, comme illustré sur la Figure 2.9

Dans une situation de non-visibilité directe, le trajet le plus fort au niveau des réponses

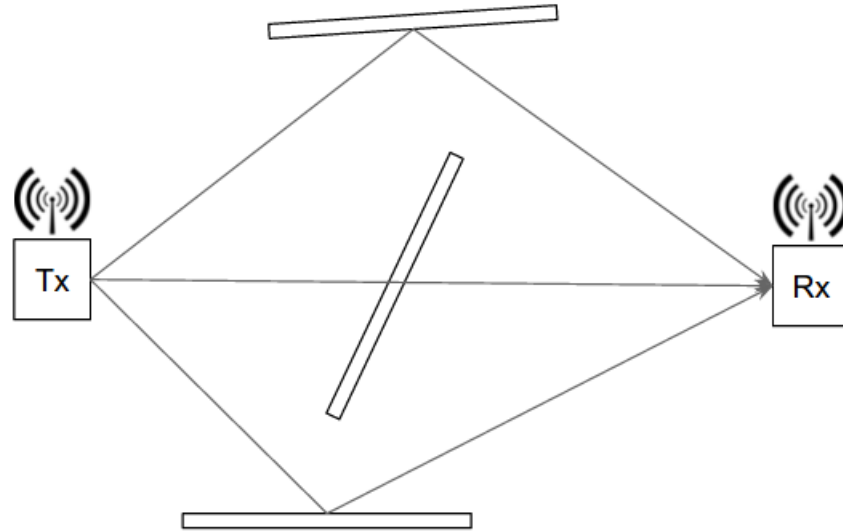


Figure 2.9 **Perte de la ligne de vue directe entre l'émetteur et le récepteur.** À cause des obstacles entre l'émetteur et le récepteur, le signal transmis a subi des modifications : des réflexions qui vont augmenter le temps de propagation, sachant qu'une erreur de 1 ns correspond à une erreur sur la distance de 30 cm. Une pénétration dans l'obstacle qui affaiblit l'amplitude du signal. Dans une telle situation, le signal qui décrit le temps de propagation réel entre l'émetteur et le récepteur aura la plus faible amplitude. Donc le récepteur peut considérer comme un signal hors ligne de vision.

impulsionnelles n'est pas nécessairement le trajet le plus court. En l'absence de toute information sur la perte de vue entre les deux équipements, une estimation précise de la distance qui les sépare est impossible. Certaines techniques ont été développées pour estimer l'instant d'arrivée du premier trajet afin d'éviter de larges erreurs sur l'estimation de la distance entre l'émetteur et le récepteur [Gezici et al. (2003) ; McGuire et al. (2003)]. L'idée consiste à effectuer à l'avance un ensemble de mesures de distance avec les nœuds de référence à des emplacements connus. Ensuite, les nouvelles mesures de distance sont comparées avec celles enregistrées afin de détecter si l'onde a subi une réflexion ou une diffraction. [Kempke et al. (2015)] présente également une solution pour s'affranchir de ce problème. Les auteurs utilisent trois antennes qui sont orientées à 0° , 120° et 240° pour permettre une couverture adéquate de la polarisation.

Dans le but d'évaluer l'impact de perte de ligne de vue sur le système proposé, deux expériences ont été réalisées. La Figure 2.10a présente des mesures de distance effectuées en ligne de vue directe entre l'émetteur et le récepteur. Nous pouvons remarquer que les résultats obtenus sont très proches des mesures de distance réelles. Par contre dans le cas de non-visibilité directe où il y a des obstacles (meubles, murs, etc) entre l'émetteur et le récepteur,

les mesures de distance sont affectées par une erreur de l'ordre de 15 cm par rapport aux mesures de distances réelles, comme le montre la Figure 2.10b.

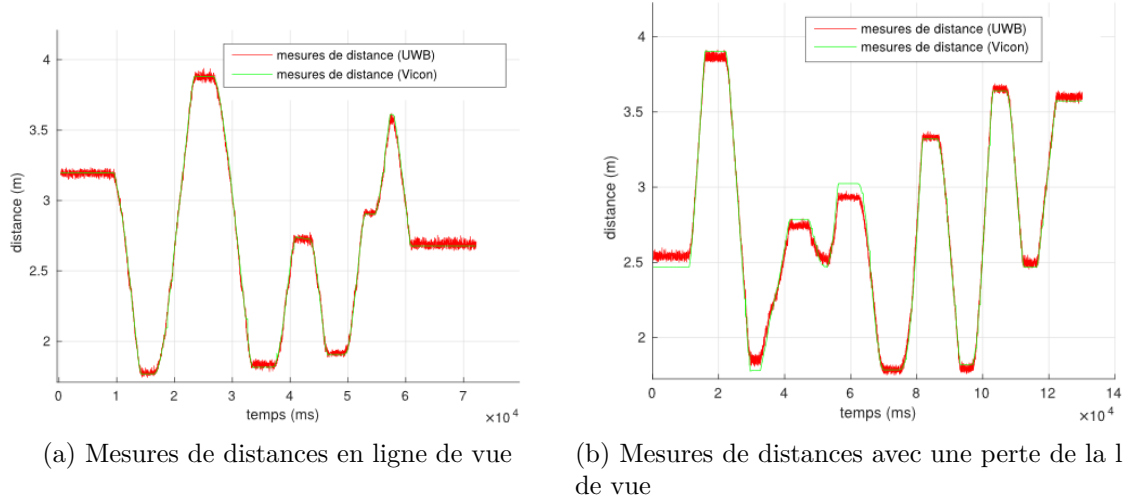


Figure 2.10 **Comparaison des mesures de distance.** La Figure (a) présente les mesures de distance effectuées en une ligne de vue directe entre les deux modules UWB. Nous pouvons remarquer que ces mesures de distance (ligne rouge) sont très proches des mesures fournies par le système Vicon (ligne verte). Par contre, sur la Figure (b) on voit qu'il y a une erreur sur les mesures de distance dans les conditions de non-visibilité directe. Dans une telle situation, l'erreur peut varier de quelques centimètres à quelques mètres en fonction du nombre d'obstacles entre l'émetteur et le récepteur.

2.2.4 Impact de la directionnalité des antennes

Non seulement la présence d'obstacles entre les modules peut causer une perte de la ligne de vue directe dans notre système, mais aussi le non-alignement des antennes de réception et d'émission peut induire des erreurs significatives pendant le calcul de la distance qui sépare les deux objets. L'alignement des antennes consiste à placer les deux modules à une orientation spécifique de façon à obtenir une intensité maximale du signal à chaque extrémité de la liaison. La Figure 2.11 présente une caractérisation de l'erreur systématique des mesures de distance en fonction de l'orientation relative des antennes des deux modules. Les expériences ont montré que le système présente une erreur minimale quand les deux antennes sont placées une en face de l'autre. Cette erreur atteint son maximum, qui est de l'ordre 15 cm, quand ils sont placés à faces opposées. Il convient également de noter que dans une telle situation les impulsions UWB circulant entre les modules UWB peuvent être réfléchies et indiquent une distance largement supérieure à la distance réelle. Ces données aberrantes peuvent être isolées

en utilisant une modélisation mathématique du comportement du mobile. Par exemple, la vitesse de déplacement du mobile nous permet d'identifier si la mesure de distance effectuée correspond à la distance réelle. La caractérisation de cette erreur a été déduite en faisant des mesures de distances entre un émetteur et un récepteur en variant l'orientation des antennes entre 0 et 2π . Avec cette caractérisation combinée avec l'estimation de l'orientation des modules fournie par une centrale inertielle, nous pouvons effectuer une compensation de l'erreur sur les mesures de distance afin de réduire l'erreur de localisation du robot.

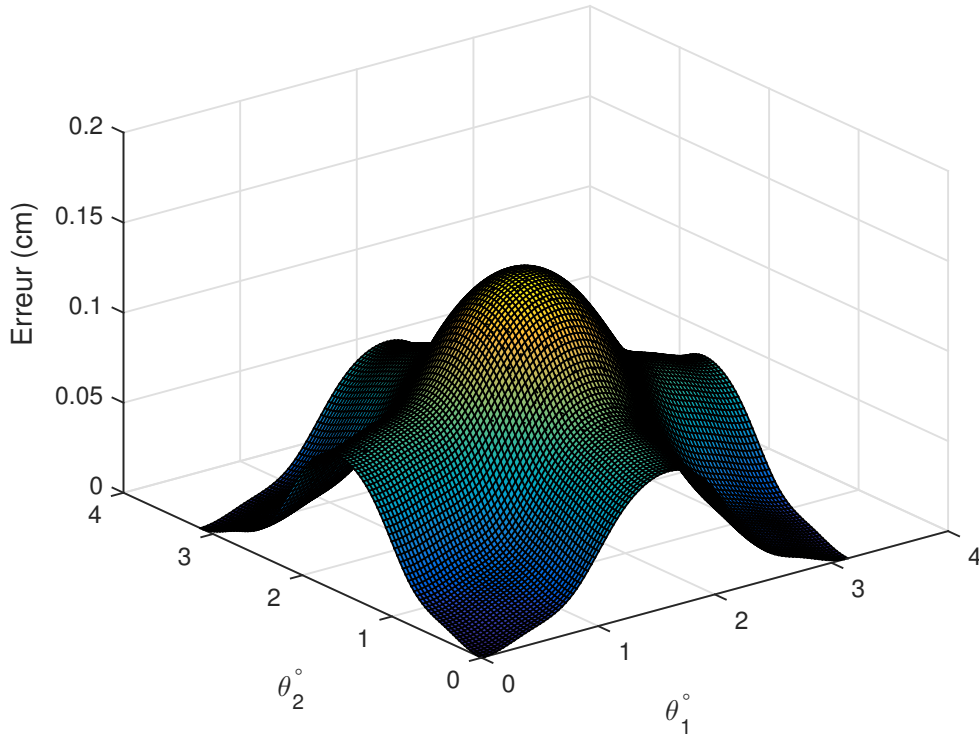


Figure 2.11 **Caractérisation de l'erreur sur les mesures de distance.** Cette caractérisation a été réalisée en utilisant deux modules UWB placés à des orientations variant de 0 à 2π pour effectuer des mesures de distance. Cette expérimentation a montré que ces mesures sont affectées par une erreur qui varie en fonction de l'angle entre les deux nœuds. Cette erreur atteint son maximum quand les modules sont placés à faces opposées.

2.3 Filtrage des mesures de distances

Filtrer les mesures de distance peut être une solution pour atténuer le bruit des mesures de distance en fonction de l'application afin d'obtenir une meilleure précision. Cependant, il y a un compromis entre le filtrage des mesures de distance et le temps de latence qui représente la

différence de temps entre la distance réelle et la distance estimée. Pour pallier cette difficulté et améliorer la vitesse de convergence du système, nous avons utilisé une version modifiée du filtre alpha-beta [Bridgewater (1978)], décrit par les équations suivantes :

$$\hat{\mathbf{x}}_k = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \hat{\mathbf{x}}_{k-1} \quad (2.1)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k + \begin{bmatrix} \alpha \\ \frac{\beta}{\delta t} \end{bmatrix} (\tilde{d}_k - \hat{d}_k), \quad (2.2)$$

Ici $\hat{\mathbf{x}}_k = [\hat{d}_k, \dot{\hat{d}}_k]^T$ représente la distance estimée d qui sépare les deux modules et sa dérivée par rapport au temps à une période k , \tilde{d}_k représente les mesures de distance brutes, δt est la période d'échantillonnage et α et β sont les coefficients du filtre à ajuster.

En pratique, nous ajustons les gains du filtre α et β en temps réel en fonction de la précision escomptée. Cela est fait grâce à un algorithme de détection de mouvement qui consiste à détecter les périodes stationnaires du nœud mobile avec un capteur inertiel. Nous utilisons pour cet algorithme un accéléromètre qui sert à mesurer les accélérations du mobile par rapport à un référentiel galiléen. Les mesures effectuées par ce capteur peuvent être exprimées avec la formule suivante :

$$\mathbf{A}_{\text{data}} = \mathbf{a} - \mathbf{g},$$

où le vecteur \mathbf{a} représente les mesures de l'accélération linéaire et \mathbf{g} représente le vecteur gravité.

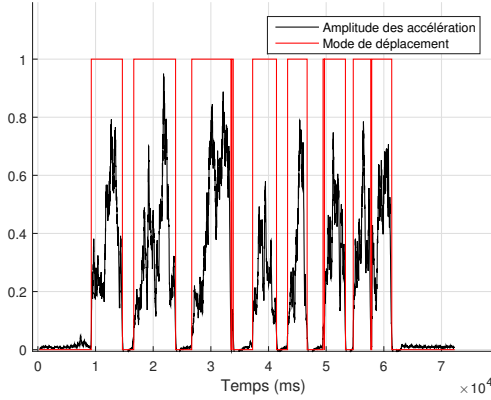
Étant donné que ce capteur peut être monté selon une orientation quelconque sur le robot, l'utilisation d'une matrice de rotation définissant le passage du référentiel galiléen au repère de navigation du robot est nécessaire. Cependant, l'amplitude du vecteur d'accélération que nous cherchons à calculer est indépendante de toute rotation du capteur. Ainsi en mode stationnaire l'accéléromètre retourne les mesures du vecteur gravité, qui s'exprime comme suit :

$$\mathbf{A}_{\text{stationary}} = -\mathbf{g}$$

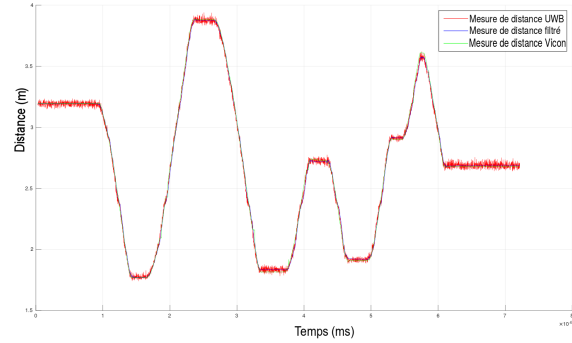
Connaissant les mesures de référence du mode stationnaire, nous pouvons calculer le vecteur d'amplitude des accélérations en faisant la différence entre les mesures retournées par l'accéléromètre \mathbf{A}_{data} et la lecture de référence $\mathbf{A}_{\text{stationary}}$, comme le montre la formule suivante :

$$\begin{aligned} |\mathbf{A}_{\text{data}} - \mathbf{A}_{\text{stationary}}| &= |(\mathbf{a} - \mathbf{g}) + \mathbf{g}| \\ &= |\mathbf{a}| > s, \end{aligned}$$

où s est un seuil ajusté pour définir le changement entre les modes “stationnaire” et “en mouvement”. Ce seuil doit être ajusté en fonction de la vitesse maximale du robot pour bien faire la différence entre ces deux modes de fonctionnement. La Figure 2.12 présente les résultats des estimations de distance obtenues en utilisant cet algorithme pour un robot se déplaçant avec une vitesse de l’ordre 1 m/s. Nous pouvons remarquer sur la Figure 2.12a que notre algorithme détecte les zones stationnaire et ajuste les gains du filtre.



(a) Détection du mouvement du robot



(b) Filtrage des mesures de distance

Figure 2.12 **Résultats obtenus de l’algorithme de filtrage des mesures de distance.** La Figure (a) présente les résultats obtenus avec l’algorithme de détection de mouvement. On voit que le vecteur d’amplitude des accélérations ne dépasse pas le seuil fixé à 0.05 quand le robot est en mode “stationnaire”. Par contre, dès que le robot commence à se déplacer l’amplitude des accélérations augmente et le robot change de mode et ajuste les gains du filtre. La Figure (b) présente les mesures de distance filtrées avec les mesures de distance fournies par le système Vicon pour évaluer la performance de l’algorithme. Nous pouvons remarquer que les mesures de distances filtrées sont moins bruitées que dans le mode “stationnaire”.

2.4 Conclusion

Dans ce chapitre, nous avons discuté de la mesure de distances par transmission UWB en nous basant sur les mesures de temps de propagation d’un signal entre deux équipements. L’approche choisie se distingue des travaux antérieurs en mettant l’accent sur l’aspect asynchrone de l’algorithme de mesure de distance et la caractérisation des sources d’erreur possible qui peuvent affecter la précision du système. En outre, nous avons discuté des compromis de précision du filtrage des mesures de distance par rapport au taux de rafraîchissement du système global, et souligné les points forts et les faiblesses de différentes configurations. Pour le protocole implémenté, nous avons pu réduire le temps de mesure d’une distance entre deux modules à 3 ms, ce qui rend ce système utilisable par plusieurs types de plate-formes

robotiques, par exemple les drones. Les évaluations des données retournées par notre algorithme de mesure des distances ont été en mesure de valider notre approche et ils ont montré que l'erreur sur les mesures de distance ne diverge pas au cours du temps, avec une erreur moyenne inférieure à 10 cm.

CHAPITRE 3 LOCALISATION UWB ET FUSION DES DONNÉES UWB/INS

Ce chapitre survole la conception d’un système de localisation UWB basé sur les mesures de distance du chapitre précédent. Nous utilisons également un système de navigation inertielle comme deuxième source d’information pour la navigation. L’idée consiste à combiner les informations provenant de différents capteurs. L’objectif de la fusion multi-capteurs est de compenser les lacunes de chacun pour produire une estimation d’état plus riche ou plus complète.

Dans ce chapitre, nous décrivons les techniques (classiques) de localisation par “ranging” utilisées pour le développement de notre système : méthode de Gauss-Newton et filtre de Kalman. Enfin, nous expliquons en détail le fonctionnement de la localisation multi-capteurs et l’estimation de l’orientation avec des modules UWB.

3.1 Localisation *UWB*

Cette section a pour but d’effectuer un panorama des solutions de localisation *UWB* utilisées dans le cadre de ce projet.

3.1.1 Méthode des moindres carrés

Un système de localisation minimal en deux dimensions comprend quatre modules dont l’un, mobile, joue le rôle de *Tag* (celui qui veut se localiser), et les trois autres, fixes, opèrent comme balises (points de référence à des positions connues), voir la Figure 3.1. Un plus grand nombre de balises permet d’obtenir un système plus précis et plus robuste. En supposant pour l’instant qu’il est possible de déterminer les distances qui séparent le *Tag* des balises, il devient possible de calculer sa position par rapport à celles-ci par des méthodes classiques, comme la méthode de Gauss-Newton pour résoudre un problème de moindres carrés [Gratton et al. (2007)].

Dénotons $\mathbf{A}_i = (x_i, y_i)$ les positions de m balises, et \tilde{d}_i la distance mesurée (et bruitée en général) entre la balise i et le *Tag*, pour $i = 1, 2, \dots, m$. Soit $\mathbf{P} = (x, y)$ la position du *Tag*. On a alors :

$$\|\mathbf{A}_i - \mathbf{P}\| \approx \tilde{d}_i, \quad i = 1, \dots, m.$$

Le seul paramètre inconnu dans cette équation est le vecteur des coordonnées du *Tag* \mathbf{P} .

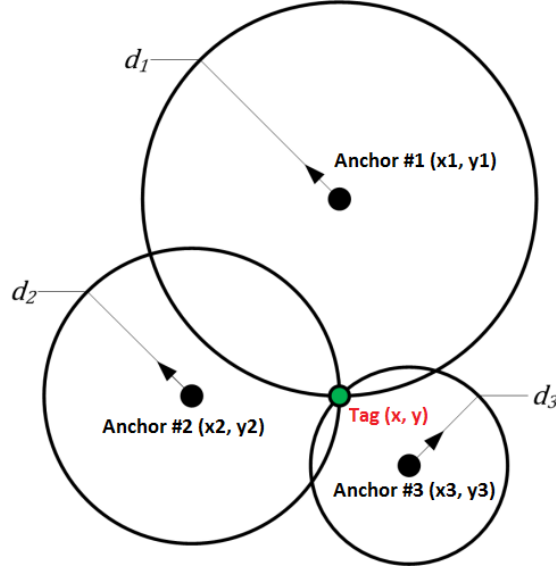


Figure 3.1 **Localisation du Tag avec trois stations de base.** Connaissant la position des stations de base et les distances qui les séparent du Tag, il est possible de tracer un cercle sur lequel il se trouve par rapport à chaque station. L'intersection d'au moins trois cercles fournit la position du Tag en deux dimensions. Le point d'intersection peut être calculé par la méthode de Gauss-Newton. La position du Tag en trois dimensions peut être estimée avec au moins quatre balises.

Nous pouvons estimer ce vecteur \mathbf{P} en résolvant le problème des moindres carrés suivant :

$$\mathbf{P} = \min_p \sum_{i=1}^m \|\mathbf{g}(\mathbf{p})\|^2 \quad (3.1)$$

avec $\mathbf{g}(\mathbf{p})$ un vecteur dont la coordonnée i représente l'erreur entre la valeur estimée et la distance mesurée \tilde{d}_i

$$g_i(\mathbf{p}) = \|\mathbf{A}_i - \mathbf{P}\| - \tilde{d}_i, \quad i = 1, 2, \dots, m \quad (3.2)$$

Pour résoudre ce problème, nous choisissons une position estimée initiale $\mathbf{P}_0 = (x_0, y_0)$ qui est améliorée de manière itérative jusqu'à ce qu'un minimum local soit trouvé, dans le cas le plus simple par un algorithme dit de Gauss-Newton, détaillé par exemple dans [Gratton et al. (2007)].

3.1.2 Filtrage de Kalman

La méthode des moindres carrés de la section précédente ne prend pas en compte de modèle du mouvement du mobile. Bien qu'il soit possible d'inclure un tel modèle dans cette méthode, une alternative est d'utiliser un filtre de Kalman, pour obtenir une solution qui permettra de mieux suivre des cibles dynamiques. Le modèle choisi pour le mobile est un modèle de marche aléatoire sur la position, qui présente moins d'oscillations de trajectoire au niveau des points de changement de direction du robot par rapport aux modèles position-vitesse *PV* et position-vitesse-accélération (*PVA*) testés [Shareef and Zhu (2009)].

Dans un espace à trois dimensions, les distances qui séparent les stations de base de l'émetteur à une position $\mathbf{P} = [x, y, z]^T$ s'expriment comme suit

$$h_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$$

avec $\mathbf{S}_i = [x_i, y_i, z_i]$ les coordonnées de la i ème station de base. Le système suivant décrit le mouvement en trois dimensions du récepteur attaché au robot

$$\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + \nu$$

où ν représente un bruit blanc de moyenne nulle et de matrice de covariance \mathbf{Q} . La matrice jacobienne des mesures de distance s'écrit :

$$\mathbf{H} = \begin{bmatrix} \frac{x-x_1}{h_1} & \frac{y-y_1}{h_1} & \frac{z-z_1}{h_1} \\ \frac{x-x_2}{h_2} & \frac{y-y_2}{h_2} & \frac{z-z_2}{h_2} \\ \vdots & \vdots & \vdots \\ \frac{x-x_n}{h_n} & \frac{y-y_n}{h_n} & \frac{z-z_n}{h_n} \end{bmatrix}$$

L'algorithme 1 récapitule le filtre de Kalman de localisation, avec \mathbf{R} la matrice de covariance des erreurs de mesure.

Initialiser $\mathbf{P} = \mathbf{P}_0$ (par ex : $[0, 0, 0]^T$) et \mathbf{M} la matrice de covariance d'erreur sur la position estimée;

while *True* **do**

```

    mesurer les  $m$  distances  $\tilde{d}_i$ ;
    calculer  $\mathbf{H}$  et  $\mathbf{h}$  avec  $h_i = \|\mathbf{S}_i - \mathbf{P}\|$ ;
     $\mathbf{K} \leftarrow \mathbf{M}\mathbf{H}^T(\mathbf{H}\mathbf{M}\mathbf{H}^T + \mathbf{R})^{-1}$ ;
     $\mathbf{P} \leftarrow \mathbf{P} + \mathbf{K}(\tilde{\mathbf{d}} - \mathbf{h})$ ;
     $\mathbf{M} \leftarrow (\mathbf{I}_{3 \times 3} - \mathbf{K}\mathbf{H})\mathbf{M}$ ;
     $\mathbf{M} \leftarrow \mathbf{M} + \mathbf{Q}$ ;

```

end

Algorithm 1: Localisation des récepteurs par filtrage de Kalman.

Performances de la localisation *UWB* par filtrage de Kalman

La Figure 3.2 présente une simulation de la localisation par filtrage de Kalman sur une trajectoire carrée d'un robot mobile, avec les trois modèles P , PV et PVA . Il convient de noter que les résultats obtenus ont été optimisés pour avoir une erreur minimale par rapport à la trajectoire réelle avec chaque modèle. Cela a été fait en évaluant les résultats obtenus par le filtre de Kalman pour une variété de valeurs de la matrice de covariance du bruit de processus \mathbf{Q} .

Afin de comparer les performances de l'algorithme entre ces trois modèles, nous avons utilisé deux métriques d'évaluation, l'erreur moyenne de distance (3.3) et l'erreur quadratique moyenne (Root Mean Square Error) définie par l'équation (3.4).

$$\text{erreur de distance} = \frac{\sum_{i=0}^n \sqrt{(\tilde{x} - x)^2 + (\tilde{y} - y)^2}}{\text{nombre des estimés}} \quad (3.3)$$

$$i_{RMSE} = \sqrt{\frac{\sum_{i=0}^n (i_{actuelle} - i_{estimée})^2}{\text{nombre des estimés}}} \quad (3.4)$$

Ici i représente l'axe x , y ou z .

Le tableau 3.1 présente les résultats de simulation obtenus pour la trajectoire carrée avec les trois modèles dans un plan en deux dimensions. Les expériences ont montré que le modèle P présente la meilleure performance de localisation par rapport aux modèles PV et PVA , en faisant moins d'oscillation au point de changement de direction du mobile. Ce modèle assume que la vitesse et l'accélération du mobile sont des bruits aléatoires, ce qui lui permet de bien suivre la trajectoire. Par contre dans les modèles PV et PVA , le filtre Kalman suit de manière plus rigide les équations cinématiques des modèles, ce qui ralentit la convergence

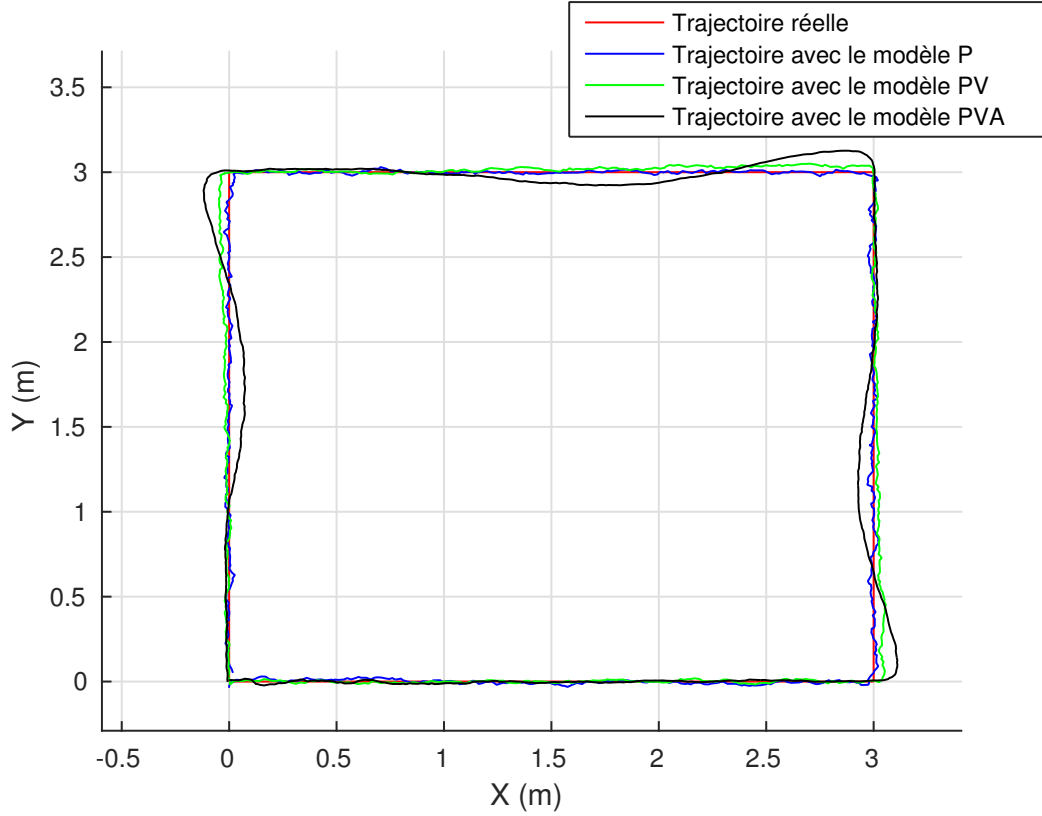


Figure 3.2 **Résultats de localisation obtenus avec l’algorithme de filtrage de Kalman sur une trajectoire carrée avec les trois modèles P , PV et PVA .** Ces résultats de simulation ont montrés que le modèle P présente moins d’oscillation au point de changement d’orientation par rapport aux deux autres modèles PV et PVA . On voit que la trajectoire noir estimé par le modèle PVA fait plus d’oscillation par rapport aux deux autres modèles.

de l’algorithme au point de changement de direction.

3.1.3 Influence de la géométrie des balises sur la localisation

Les positions des balises dans la zone d’exploration ont une influence déterminante sur la qualité de la localisation. Les expériences ont montré que l’erreur sur les mesures de distance est amplifiée dans certaines conditions, ce qui affecte considérablement l’estimation de position du mobile. La Figure 3.3b présente une situation dans laquelle suffisamment de balises sont disponibles pour localiser le mobile dans un plan en deux dimensions, sans que pour autant une bonne estimation puisse être effectuée. Nous pouvons remarquer que l’emplacement du robot est entaché d’une large incertitude. Dans une telle situation, même sans tenir en compte de l’erreur des mesures de distance UWB, il est difficile de déterminer précisément la position du point de tangence entre les trois cercles noirs décrivant l’emplacement réel du

Tableau 3.1 Taux d'erreur sur l'estimation de position par filtrage de Kalman avec les modèles P , PV et PVA

Modèles	Erreur de distance (cm)	x_{RMSE} (cm)
P	8.7	7.2
PV	8.85	7.9
PVA	9.2	8.94

robot.

La Figure 3.3a présente une façon possible d'établir une bonne géométrie en présence des erreurs sur les mesures de distance. Nous pouvons remarquer que les cercles noirs s'intersectent exactement en un seul point et donnent une seule solution au problème de localisation. Les cercles rouges, qui représentent les mesures de distance UWB, s'intersectent dans une zone très proche de l'emplacement réel du *Tag*, ce qui permet à l'algorithme de localisation de converger vers une solution avec un taux d'erreur relativement faible.

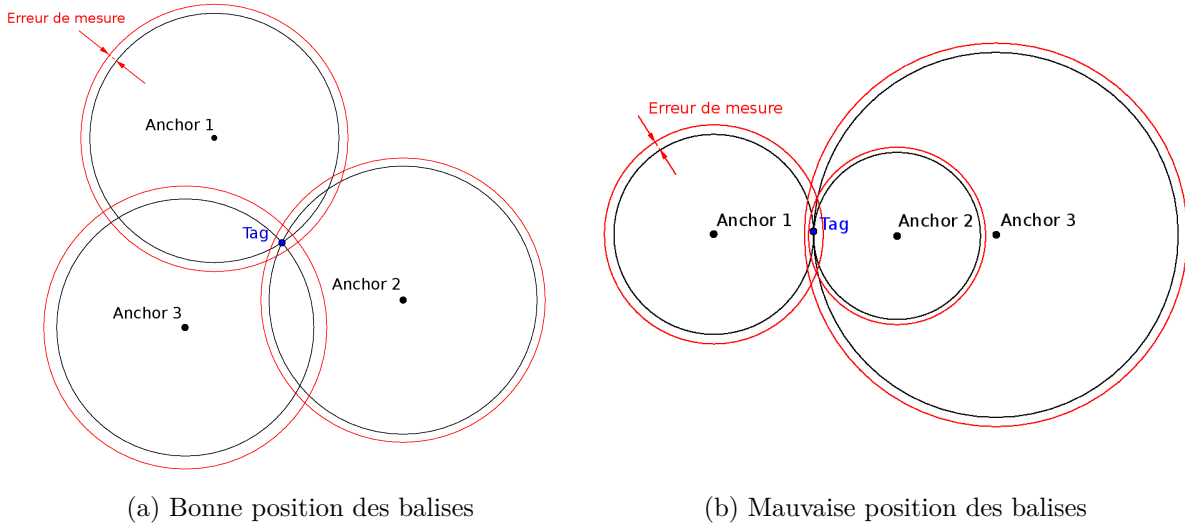


Figure 3.3 **Influence de la géométrie des balises sur la précision de localisation.** Dans la configuration de la Figure (a), on voit bien que les trois cercles noirs s'intersectent en un seul point. Dans une telle situation, l'algorithme de localisation converge bien vers une solution très proche de la position réelle. Par contre, nous pouvons remarquer qu'il est difficile de repérer le point de tangence entre les trois cercles. Ainsi l'estimation de la position du mobile sera entachée d'une large incertitude.

Un critère pour déterminer si la configuration est favorable à une bonne estimation de la position du mobile est le coefficient de dilution de précision géométrique (*Geometric Dilution*

Of Precision ou *GDOP*) [Yarlagadda et al. (2000)]. La définition mathématique du calcul de coefficient de dilution de précision est décrite par les équations suivantes :

$$J = \begin{bmatrix} \frac{\partial g_1(x)}{\partial x} \\ \vdots \\ \frac{\partial g_n(x)}{\partial x} \end{bmatrix} \quad (3.5)$$

$$G(\hat{x}) = (J(\hat{x})^T J(\hat{x}))^{-1} \quad (3.6)$$

$$GDOP = \sqrt{\text{Tr}(G)} \quad (3.7)$$

$$VDOP = \sqrt{G_{33}} \quad (3.8)$$

$$HDOP = \sqrt{G_{11} + G_{22}} \quad (3.9)$$

Ici $g_i(x)$ définit par l'équation (3.2), $G(\hat{x})$ est une amplification de l'erreur dans l'estimation de position, qui dépend de la géométrie des stations de base par rapport à la position du mobile. $GDOP$ est le facteur de dilution de précision et $VDOP$ et $HDOP$ représentent la qualité du positionnement vertical et horizontal. Un coefficient GDOP petit indique une bonne géométrie des balises pour la localisation :

- Une valeur proche de 1 correspond à une faible amplification des erreurs de mesures de distance dans l'algorithme de localisation.
- Un indice entre 2 et 4 est considéré comme assez précis pour une application de localisation qui nécessite une précision élevée.
- Une valeur entre 4 et 8 représente la limite minimale pour garantir la convergence de l'algorithme de localisation vers une position proche de la position réelle.
- Au-delà de 20 les estimations de position sont très imprécises et fortement entachées d'incertitude dans la majorité des cas. Dans une telle situation le système ne peut pas être utilisé pour des applications de navigation robotique.

La Figure 3.4 présente les résultats de simulation obtenus pour le GDOP en fonction de l'emplacement du robot. Dans ce scénario de simulation, le robot parcourt une surface de 9 m². Il mesure les distances qui le séparent de quatre balises placées respectivement aux coordonnées suivantes (0;0;2.5), (0;3;2.5), (3;3;2.5) et (3;0;2.5). La géométrie des balises utilisée dans ce scénario ne garantit pas une couverture de toute la zone d'exploration. Ainsi, quand le robot se déplace dans la zone de couverture c'est-à-dire à l'intérieur de la formation des points de référence comme décrit par la Figure 3.3a, nous pouvons remarquer que le coefficient de dilution de précision indique une valeur très proche de 1. Dans cette partie de la zone de déplacement, nous obtenons le minimum d'erreur de localisation possible qui peut affecter notre système. Tandis que quand le robot sort de la formation l'indice de dilution de précision indique une valeur qui croît en fonction de la distance qui le sépare de la formation

des points de référence.

Cette simulation nous permet d'anticiper le taux d'erreur qui peut affecter notre système de localisation à cause d'un mauvais emplacement des balises dans la zone d'exploration. Elle nous permet également de bien choisir le nombre de balises ainsi que leur position afin de minimiser l'influence de la géométrie des balises sur la localisation.

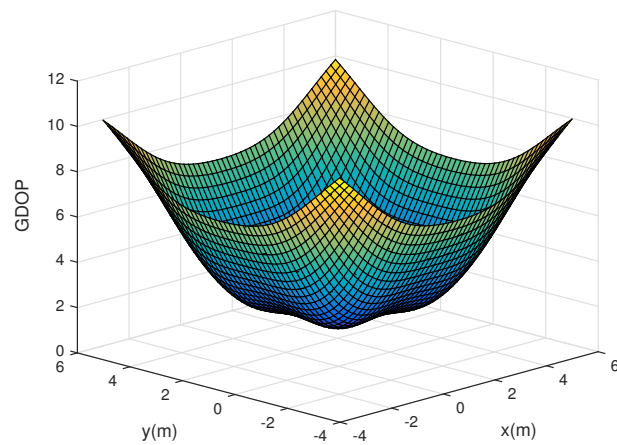


Figure 3.4 Coefficient de dilution de précision en fonction de la géométrie des balises.

3.2 Localisation multi-capteurs

Dans cette section, nous développons tout d'abord la théorie mathématique du système de fusion de données. Ensuite, nous présentons l'algorithme développé pour l'estimation de l'orientation avec l'*IMU*. Puis, nous expliquons en détail la conception du filtre de Kalman pour la fusion de données. Enfin, nous concluons en évaluant les performances de notre solution et en discutant des résultats obtenus.

3.2.1 Système de navigation inertielle

Un système de navigation inertielle est composé des capteurs d'accélération et de rotation afin de déterminer le mouvement absolu d'un mobile. Ce système de navigation peut être divisé en deux parties : la partie matérielle qui comprend l'unité de navigation inertielle (*Inertial Measurement Unit* ou *IMU*), et le calculateur responsable de l'estimation de l'état du mobile, qui intègre les équations de navigation alimentées par les capteurs, comme le montre la Figure 3.5.

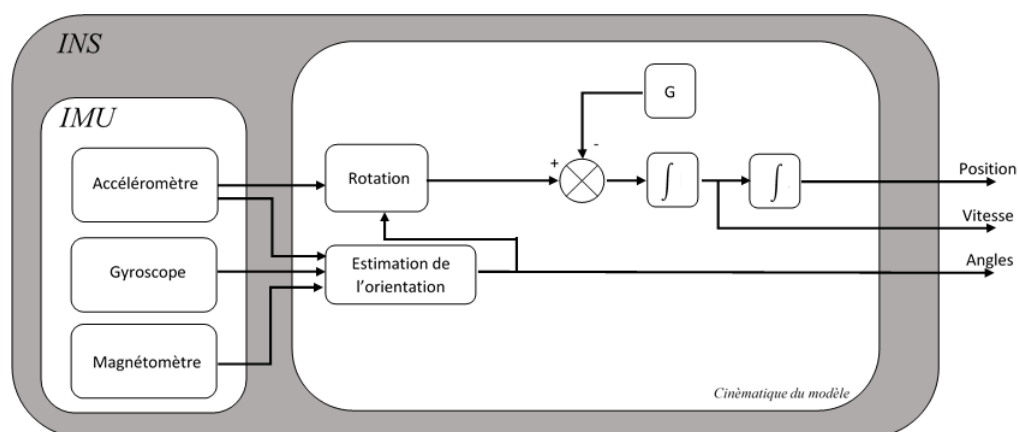


Figure 3.5 Système de navigation inertielle (INS) [Borràs Sillero (2012)].

Description de la partie matérielle d'un système de navigation inertielle

L'unité de navigation inertielle peut fournir plusieurs types d'information décrivant la nature du mouvement effectué par un mobile. Cette unité est composée de trois capteurs :

- **Un accéléromètre à 3-axes** qui mesure les accélérations du mobile par rapport à un référentiel galiléen.
- **Un gyroscope à 3-axes** qui donne la vitesse angulaire du repère mobile par rapport à un référentiel galiléen.

- **Un magnétomètre à 3-axes** qui sert à mesurer la direction du champ magnétique terrestre.

Il existe différentes configurations pour le montage de ces capteurs formant l'*IMU* qui sont généralement classés dans deux catégories : les centrales à plate forme stabilisée et les centrales inertielles liées au bâti du robot ou “strap-down”. Dans une centrale “strap-down”, l'IMU est montée sur le châssis du mobile. Dans une telle configuration, les mesures d'accélération et de la vitesse angulaire sont exprimées dans le repère mobile, le traitement des données permet ensuite de faire les changements de base pour le passage du repère mobile au repère de navigation. Les avantages de cette approche sont la réduction de prix et de la complexité mécanique de la réalisation du système, ainsi que la consommation d'énergie. Par contre dans une centrale à plate forme stabilisée, l'accéléromètre est stabilisé par le gyroscope avec des rotations mécaniques du châssis, afin de maintenir l'alignement du repère mobile avec le repère de navigation. Cette configuration donne de meilleurs résultats, mais nécessite une construction rigoureuse et des précautions particulières afin de minimiser les frottements.

Les mesures de l'*IMU* sont généralement affectées par plusieurs sources d'erreur comme un biais systématique et le bruit des mesures. Ces facteurs engendrent des dérives non négligeables sur l'estimation de la posture du robot au fil du temps. Afin de limiter l'impact des dérives sur la précision du système, une phase d'étalonnage des capteurs est nécessaire [Benjamin (2013)]. Ces capteurs peuvent être sensibles aussi à d'autres facteurs comme la température et les vibrations durant le déplacement.

Estimateur de position et de l'orientation avec l'*IMU*

Avec un capteur comme l'accéléromètre qui mesure l'accélération d'un mobile, il serait possible de calculer la variation de la vitesse \mathbf{V}^n au cours du déplacement du robot ainsi que sa position \mathbf{P}^n en effectuant des intégrations successives de l'accélération en fonction du temps, comme le montrent les équations (3.10) et (3.11).

$$\dot{\mathbf{P}} = \mathbf{V} \quad (3.10)$$

$$\dot{\mathbf{V}} = \mathbf{a} \quad (3.11)$$

Le capteur gyroscope peut être également utilisé pour déterminer l'orientation de la centrale inertielle montée sur le robot mobile. Avec cette information, il est possible d'exprimer les accélérations dans le repère de navigation n avant de procéder à l'intégration. Ce changement de base du repère mobile b vers le repère de navigation n se fait par la matrice de rotation \mathbf{R}_n^b (3.12). De cette façon, l'*IMU* peut donner une estimation de la posture du robot dans le

repère de navigation, comme le montre la Figure 3.6.

$$\begin{bmatrix} a_x^n \\ a_y^n \\ a_z^n \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} \quad (3.12)$$

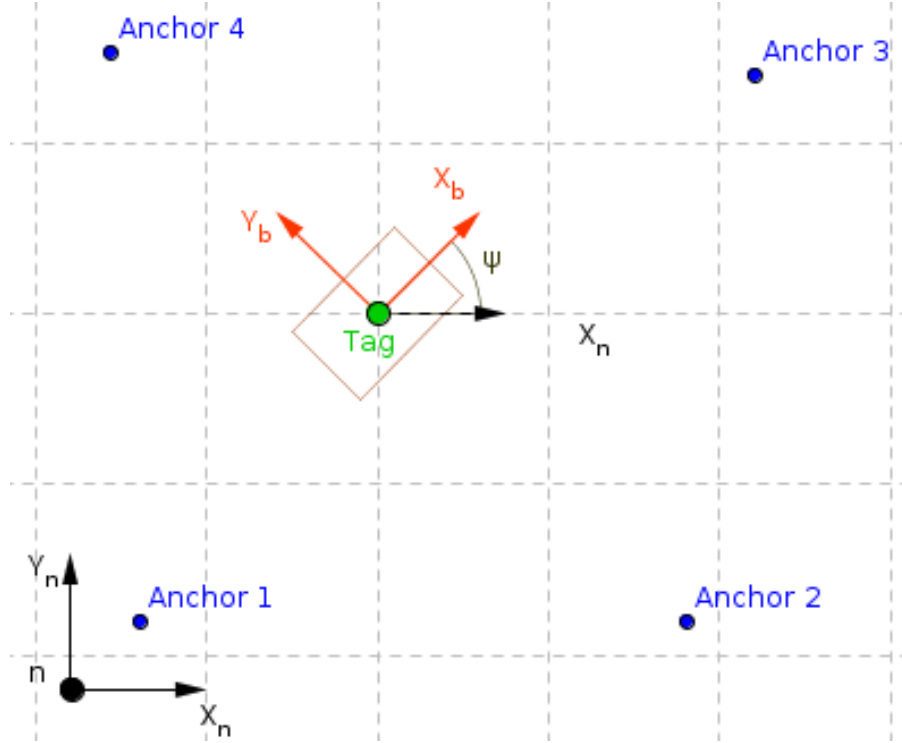


Figure 3.6 **Repère de navigation du robot**. Le repère n représente le repère de navigation du robot. L'emplacement du robot ainsi que les coordonnées des stations de base *Anchor* 1, 2, 3 et 4 sont exprimés dans ce repère. Les mesures des accélérations et des vitesses angulaires fournies par l'IMU sont exprimées dans le repère mobile b lié au robot.

Équations de navigation : Dans le but d'estimer le vecteur d'état de la navigation composé de la position \mathbf{P} , la vitesse \mathbf{V} et l'orientation Θ du mobile, nous utilisons une discrétisation des équations de navigation (3.10) et (3.11) qui prennent la forme du système d'équations (3.13), et qui supposent que les mesures provenant des différents capteurs sont

constantes sur une période d'échantillonnage.

$$\begin{bmatrix} \mathbf{P}_{k+1}^t \\ \mathbf{V}_{k+1}^t \\ \boldsymbol{\Theta}_{k+1}^t \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \delta t \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{P}_k^t \\ \mathbf{V}_k^t \\ \boldsymbol{\Theta}_k^t \end{bmatrix} + \begin{bmatrix} \frac{\delta t^2}{2} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \delta t \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \delta t \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{R}_b^n \\ \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{a}}_k^b \\ \tilde{\boldsymbol{\omega}}_k^b \end{bmatrix}, \quad (3.13)$$

Ici δt est la période d'échantillonnage de l'*IMU*, \mathbf{R}_b^n représente la matrice de rotation du repère mobile b par rapport au repère de navigation n et $\tilde{\mathbf{a}}^n$ et $\tilde{\boldsymbol{\omega}}^n$ représentent respectivement le vecteur d'accélération et les données du gyroscope dans le repère mobile.

3.2.2 Estimation de l'orientation avec l'*IMU*

Le système est conçu à base des cartes de développement **STM32f3DISCOVERY**¹ équipées d'un micro-contrôleur **STM32F303VC**, d'un gyroscope, d'un magnétomètre et d'un accéléromètre 3-axes. Cependant, ces capteurs inertiels ne sont pas de très bonne qualité et fournissent des mesures entachées d'incertitude. Par conséquent, une calibration de ces mesures est nécessaire pour éliminer la dérive des mesures et avoir une bonne estimation de l'attitude du robot [Benjamin (2013)]. Dans ce travail, nous avons choisi d'utiliser les quaternions pour la fusion des données des capteurs inertiels. Cette représentation est la forme la plus simple à programmer et la plus efficace en termes de temps de calcul. Elle permet également de s'affranchir du problème de singularité avec les angles d'Euler. L'intégration des données de tous ces capteurs a été faite en utilisant un algorithme de descente de gradient décrit dans [Madgwick et al. (2011)]. La Figure 3.7 illustre l'algorithme complet de l'estimation de l'orientation. Elle décrit également la méthode de la compensation de la dérive du gyroscope avec les données de l'accéléromètre et du magnétomètre.

Les résultats obtenus de l'estimation de l'orientation du mobile avec la méthode de descente de gradient sont très satisfaisants étant donné qu'une simple intégration des vitesses angulaires nous donne des résultats qui manquent de précision. La Figure 3.8 nous présente une comparaison des résultats de l'estimation des trois angles d'Euler (ϕ , θ et ψ) pour une centrale inertielle au repos ($0^\circ, 0^\circ, 170^\circ$). Ces trois angles ϕ , θ et ψ décrivent respectivement des rotations successives autour de l'axe z , l'axe x et l'axe y du repère mobile par rapport au repère de navigation.

1. <http://www.st.com/web/en/catalog/tools/FM116/SC959/SS1532/PF254044>

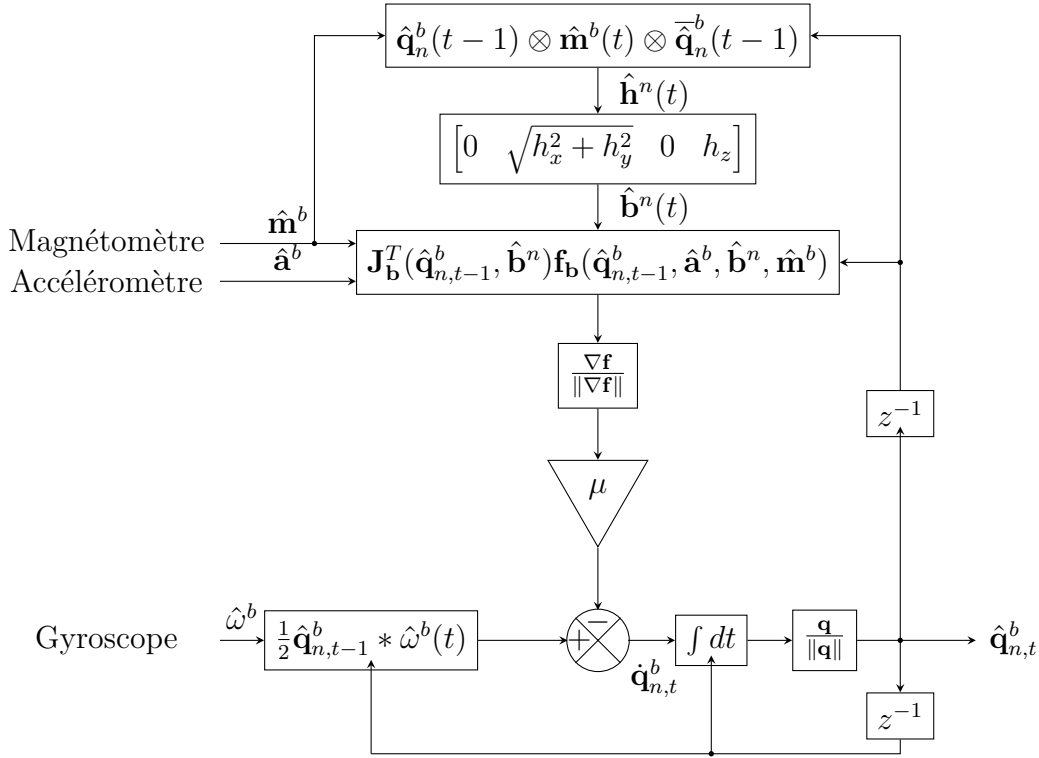


Figure 3.7 **Algorithme de descente de gradient** [Madgwick et al. (2011)] : $\hat{\omega}^b(t)$ décrit le quaternion contenant les mesures de vitesse angulaire sur les axes x , y et z fournis par le capteur gyroscope. Ces mesures sont exprimées dans le repère mobile b . Le passage du repère mobile b au repère de navigation n se fait par le quaternion $\hat{\mathbf{q}}_{n,t-1}^b$. Pour la compensation des dérives des mesures de vitesse angulaire, nous utilisons les données fournies par les deux capteurs magnétomètre $\hat{\mathbf{m}}^b$ et accéléromètre $\hat{\mathbf{a}}^b$. $\hat{\mathbf{h}}^n(t)$ décrit le vecteur champ magnétique dans le repère de navigation n et $\hat{\mathbf{b}}^n(t)$ représente la composition du problème d'inclinaison des magnétomètres. La compensation de l'erreur sur l'estimation d'angle fourni par le gyroscope se fait par l'algorithme de descente de gradient décrit par la formule suivante : $\mathbf{J}_{\mathbf{b}}^T(\hat{\mathbf{q}}_{n,t-1}^b, \hat{\mathbf{b}}^n) \mathbf{f}_{\mathbf{b}}(\hat{\mathbf{q}}_{n,t-1}^b, \hat{\mathbf{a}}^b, \hat{\mathbf{b}}^n, \hat{\mathbf{m}}^b)$. Généralement, cette fonction nécessite plusieurs itérations à chaque nouvelle mesure de l'orientation du mobile pour converger. Cependant, l'algorithme peut donner de bons résultats en l'exécutant itération par itération à condition que le taux de convergence de l'orientation estimée μ soit supérieur ou égal à la vitesse de changement d'orientation du mobile. Les auteurs de [Madgwick et al. (2011)] présentent une méthode d'estimation de la valeur optimale de μ décrit par l'équation (25).

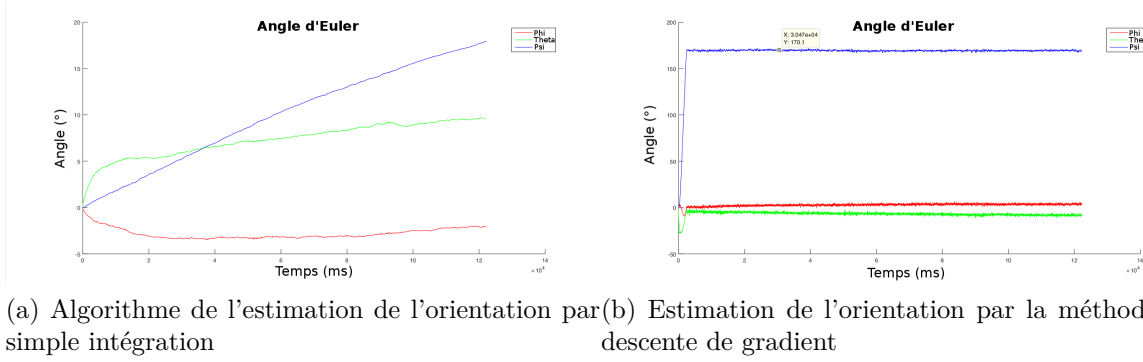


Figure 3.8 **Résultats d'estimation de l'orientation.** La méthode de descente de gradient présente de meilleurs résultats par rapport à la méthode d'intégration simple des données de l'IMU. Nous pouvons remarquer sur la Figure (b) que l'algorithme converge beaucoup plus rapidement vers l'orientation réelle du robot. De plus, il assure une meilleure compensation de la dérive des mesures de vitesse angulaire fournie par le gyroscope.

3.2.3 Filtre de Kalman Étendu pour la fusion de données

L'algorithme de l'intégration des capteurs inertiels avec les mesures de distance UWB a été développé avec un filtre de Kalman étendu (EKF), comme décrit dans [Borràs Sillero (2012)]. La Figure 3.9 présente l'architecture générale de l'algorithme, qui s'exécute en deux phases distinctes : prédiction et mise à jour. La phase de prédiction consiste à estimer l'état du système par la propagation des équations de navigation (3.13) alimentées par les mesures de la centrale inertielle afin de produire une estimation de l'état courant. Dans l'étape de mise à jour, les dernières mesures de distance *UWB* sont utilisées pour corriger l'état prédit dans le but d'améliorer les estimations de l'*INS*.

Le vecteur d'état du filtre EKF (3.14) représente l'erreur dans le vecteur de navigation. Les erreurs à estimer sont les erreurs de la position \mathbf{P} , la vitesse \mathbf{V} et le biais de l'accéléromètre $\mathbf{b}\mathbf{a}^b$. Il est important de noter que les erreurs dans les éléments de navigation sont exprimées directement dans le repère n de la navigation, étant donné qu'elles sont corrigées après avoir propagé la dynamique inertielle. Par contre, les biais des mesures de l'accélération sont représentés dans le repère b lié au robot, car la correction s'effectue avant la propagation des équations de navigation.

$$\delta \mathbf{X}_k = \begin{bmatrix} \delta \mathbf{P}^n(k) \\ \delta \mathbf{V}^n(k) \\ \mathbf{b}\mathbf{a}^b(k) \end{bmatrix} \quad (3.14)$$

Notons aussi que le symbole δ utilisé dans la définition du vecteur d'état de ce filtre n'a pas

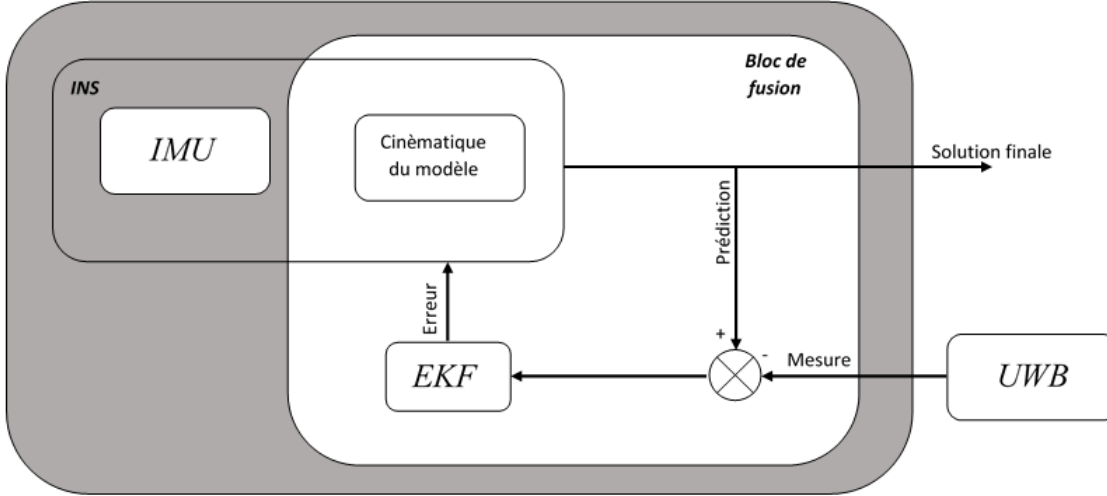


Figure 3.9 Architecture du système de fusion de données [Borràs Sillero (2012)].

de relation directe avec celui du vecteur d'état de l'INS. C'est pourquoi il n'y a pas forcément le même nombre d'éléments d'état.

À ce stade, nous pouvons définir les équations de l'algorithme associé à chaque phase du filtre EKF. Dans la phase de prédiction, l'algorithme commence par la propagation de l'équation (3.15). La matrice \mathbf{E}_k (3.16) représente la matrice d'erreur reliant l'erreur dans le vecteur de navigation à l'instant $k - 1$ à celui de l'état actuelle à l'instant k , voir page 43 de [Nilsson et al. (2009)] et [Farrell (2008)].

$$\delta \hat{\mathbf{X}}_k^- = \mathbf{E}_k \delta \hat{\mathbf{X}}_{k-1} \quad (3.15)$$

$$\mathbf{E}_k = \begin{bmatrix} \mathbf{I}_{3 \times 3} & T \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & T \hat{\mathbf{R}}_b^n \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (3.16)$$

L'équation (3.17) représente la matrice de covariance à priori :

$$\Sigma_k^- = \mathbf{E}_k \Sigma_{k-1} \mathbf{E}_k^T + \mathbf{Q}_k \quad (3.17)$$

Ici \mathbf{Q}_k est la matrice de covariance du bruit du modèle. La valeur de cette matrice doit être ajustée en fonction de la vitesse de convergence escomptée de l'algorithme.

La phase de mise à jour est alimentée par la solution de localisation fournie par la centrale inertielle et les mesures de distance *UWB*. Le vecteur des mesures de distance $h(\delta\hat{\mathbf{X}}_k^-)$ ci-dessous correspond à la différence entre les estimations de positions provenant de l'INS et les positions connues des balises $\mathbf{S}_i(x, y, z)$.

$$\mathbf{h}(\delta\hat{\mathbf{X}}_k^-) = \begin{bmatrix} \|\mathbf{S}_1 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\| \\ \vdots \\ \|\mathbf{S}_4 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\| \end{bmatrix} \quad (3.18)$$

À partir de l'équation (3.18), nous pouvons alors définir la matrice jacobienne

$$\begin{aligned} \mathbf{H}_k &= \frac{\delta(\mathbf{h}(\delta\hat{\mathbf{X}}_k^-))}{\delta(\delta\hat{\mathbf{X}}_k^-)} \\ &= \begin{bmatrix} \frac{S_{x,1} - (\hat{x}_k + \delta\hat{x}_k^-)}{\|\mathbf{S}_1 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\|} & \frac{S_{y,1} - (\hat{y}_k + \delta\hat{y}_k^-)}{\|\mathbf{S}_1 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\|} & \frac{S_{z,1} - (\hat{z}_k + \delta\hat{z}_k^-)}{\|\mathbf{S}_1 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\|} & \mathbf{0}_{1 \times 6} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{S_{x,4} - (\hat{x}_k + \delta\hat{x}_k^-)}{\|\mathbf{S}_4 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\|} & \frac{S_{y,4} - (\hat{y}_k + \delta\hat{y}_k^-)}{\|\mathbf{S}_4 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\|} & \frac{S_{z,4} - (\hat{z}_k + \delta\hat{z}_k^-)}{\|\mathbf{S}_4 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\|} & \mathbf{0}_{1 \times 6} \end{bmatrix} \end{aligned} \quad (3.19)$$

Une fois la matrice jacobienne calculée, le gain du filtre EKF peut être calculé avec (3.20). Ensuite, l'erreur de la solution fournie par la centrale inertielle est corrigée par (3.21), ainsi que la matrice de covariance a posteriori exprimée par (3.22).

$$\mathbf{K}_k = \Sigma_k^- \mathbf{H}_k^T [\mathbf{H}_k \Sigma_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1} \quad (3.20)$$

$$\delta\hat{\mathbf{X}}_k = \delta\hat{\mathbf{X}}_k^- + \mathbf{K}_k (\delta\mathbf{z}_k - \mathbf{h}(\delta\hat{\mathbf{X}}_k^-)) \quad (3.21)$$

$$\Sigma_k = [\mathbf{I}_{9 \times 9} - \mathbf{K}_k \mathbf{H}_k] \Sigma_k^- \quad (3.22)$$

L'algorithme 2 décrit la séquence d'exécution de ces équations qui se déroule sur deux étapes : une prédiction de l'état du robot avec l'INS et une phase de correction avec les mesures de distance *UWB*.

La Figure 3.10 montre une représentation tridimensionnelle de la convergence de l'algorithme de localisation vers la position du robot placée à une position fixe avec l'orientation de $(0^\circ, 0^\circ, 10^\circ)$,

```

initialization;
while data IMU available do
     $\tilde{\mathbf{u}}_k = \text{read}(\text{accelerometer}, \text{gyroscope});$ 
     $f_{ins\_orientation}(\tilde{\mathbf{u}}_k);$ 
    Compute  $E_k$ ;
    if data UWB available then
         $\mathbf{K}_k \leftarrow \Sigma_k^- \mathbf{H}_k^T [\mathbf{H}_k \Sigma_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1};$ 
         $\delta \hat{\mathbf{X}}_k \leftarrow \mathbf{K}_k (\delta \mathbf{z}_k - \mathbf{h}(\delta \hat{\mathbf{X}}_k^-));$ 
         $\Sigma_k \leftarrow [\mathbf{I}_{9 \times 9} - \mathbf{K}_k \mathbf{H}_k] \Sigma_k^-;$ 
         $\hat{\mathbf{X}}_k(1:6) \leftarrow \hat{\mathbf{X}}_k(1:6) + \delta \hat{\mathbf{X}}_k(1:6);$ 
         $\delta \hat{\mathbf{X}}_k(1:6) \leftarrow 0;$ 
    end
     $\hat{\mathbf{X}}_{k+1} \leftarrow f_{ins}(\hat{\mathbf{X}}_k, \tilde{\mathbf{u}}_k);$ 
     $\Sigma_{k+1} \leftarrow \mathbf{E}_k \Sigma_k \mathbf{E}_k^T + \mathbf{Q}_k;$ 
end

```

Algorithm 2: Algorithme de la fusion des données des capteurs : La boucle rapide de cet algorithme qui fournit une prédiction de l'état du robot s'exécute avec la fréquence de l'*IMU*. La fonction $f_{ins_orientation}$ nous donne une estimation de l'attitude du robot en utilisant l'algorithme de descente de gradient, voir Figure 3.7. Lorsque les mesures de distance deviennent disponibles, elles sont sauvegardées dans le vecteur $\delta \mathbf{z}_k$. Ensuite, l'erreur dans le vecteur d'état est corrigée par la différence du vecteur $\delta \mathbf{z}_k$ et de la fonction de prédiction des mesures $\mathbf{h}(\delta \hat{\mathbf{X}}_k^-)$. Après la correction de l'état, l'erreur a priori est réinitialisée à zéro. En outre, le biais de l'accélération est propagé à chaque itération et les valeurs obtenues sont utilisées dans la correction des mesures de l'*IMU*. Notons aussi que le changement dans le biais est modélisé comme une marche aléatoire.

3.2.4 Estimation de l'orientation par mesures *UWB* multiples

Le système *UWB* avec un seul nœud mobile n'est pas en mesure de fournir des informations d'attitude sur le mobile. Une façon possible d'estimer l'orientation avec les modules *UWB* est d'utiliser plusieurs *Tags* sur le robot. À cet effet, nous avons modélisé trois nœuds *UWB* attachés au mobile formant un triangle équilatéral. Ces trois modules communiquent avec quatre stations de base pour l'estimation de leurs positions. Connaissant les positions des émetteurs sur le robot et avec les coordonnées de ces derniers dans le repère de navigation, nous déterminons la matrice de rotation \mathbf{R}_b^n et donc l'attitude du robot (ϕ_{UWB} , θ_{UWB} et ψ_{UWB}), comme le montrent les équations suivantes

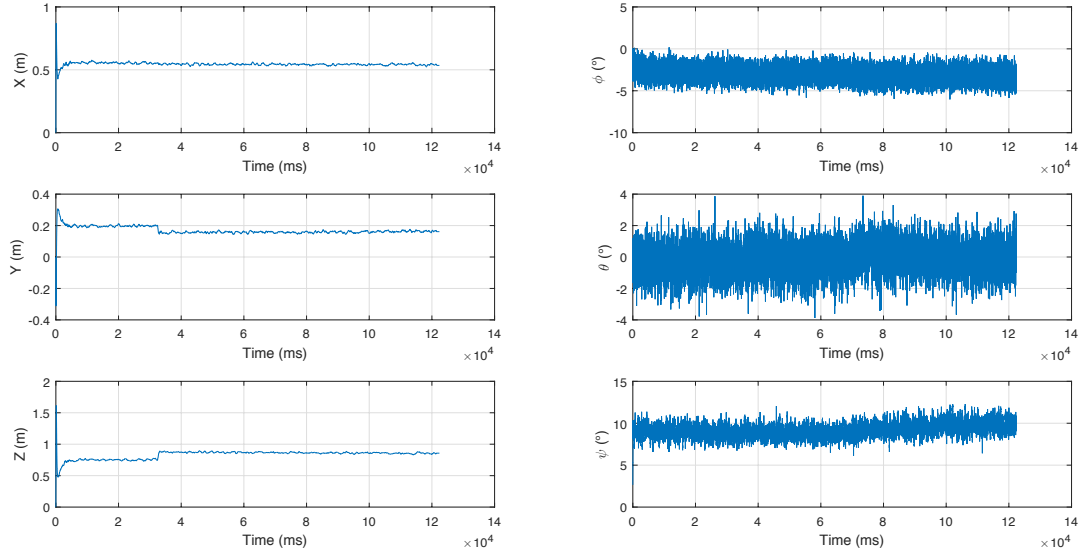


Figure 3.10 **Résultats de simulation du système de fusion de données UWB/INS.** Cette simulation a été réalisée avec les données réelles fournies par un module UWB et un IMU placés à une position fixe au centre de la formation géométrique des balises. Nous pouvons remarquer que l'algorithme converge vers une solution très proche de l'emplacement réel.

$$\begin{aligned}
 \mathbf{Pos}_n &= \mathbf{R}_b^n \mathbf{Pos}_b \\
 \mathbf{R}_b^n &= \mathbf{Pos}_n \mathbf{Pos}_b^\dagger \\
 &= \mathbf{Pos}_n \mathbf{Pos}_b^T (\mathbf{Pos}_b \mathbf{Pos}_b^T)^{-1} \\
 \phi_{UWB} &= \text{atan2}(R_b^n(2, 1), R_b^n(2, 2)) \\
 \theta_{UWB} &= -\sin(R_b^n((2, 1))) \\
 \psi_{UWB} &= \text{atan2}(R_b^n(1, 0), R_b^n(0, 0))
 \end{aligned} \tag{3.23}$$

Ici \mathbf{Pos}_n et \mathbf{Pos}_b représentent les coordonnées des *Tag* dans le repère de navigation n et le repère mobile b .

L'emplacement des modules *UWB* sur le robot affecte considérablement la précision de l'estimation de l'orientation. En effet, il y a un compromis entre la précision des estimées et la longueur des côtés du triangle. Nous pouvons remarquer sur la Figure 3.11 que l'erreur varie de quelques degrés en fonction de la longueur des côtés du triangle d .

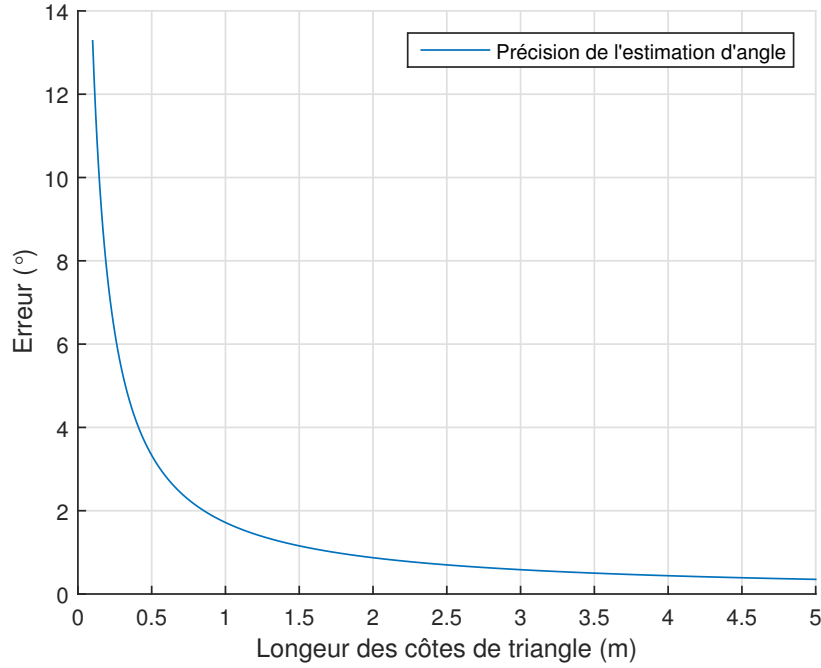


Figure 3.11 **L'erreur sur l'estimation de l'orientation *UWB* en fonction de la longueur des côtés du triangle d .** La précision de l'orientation *UWB* dépend cruellement de l'erreur sur les mesures de distance *UWB*. Dans cette caractérisation, l'erreur des mesures de distance a été fixée à une valeur de 10 cm, qui représente l'erreur maximale qu'on peut obtenir dans des conditions de visibilité directe entre les différents modules. Nous pouvons remarquer que l'estimation de l'orientation diverge quand d est inférieure à 10 cm. Pour une valeur de d supérieure à 10 cm, l'algorithme donne une estimation d'orientation de l'ordre quelques degrés.

Les modifications apportées sur l'algorithme de localisation s'appliquent en premier lieu à la matrice d'observation du filtre de Kalman étendu qui tient compte maintenant de l'estimation de l'orientation *UWB*. Le vecteur (3.24) décrit le nouveau vecteur d'état du filtre EKF.

$$\delta \mathbf{X}_k = \begin{bmatrix} \delta \mathbf{P}^n(k) \\ \delta \mathbf{v}^n(k) \\ \delta \Theta(k) \\ b \mathbf{a}^b(k) \end{bmatrix} \quad (3.24)$$

Ici $\delta \Theta = [\delta \phi \quad \delta \theta \quad \delta \psi]^T$ représente l'erreur sur les angles d'Euler.

Dans la phase de prédiction, l'algorithme estime le changement de l'orientation avec la centrale inertielle en calculant le changement de l'orientation effectué par le robot à chaque

nouvelle mesure. Aussitôt que des mesures de distance sont disponible, l'algorithme les fusionne avec l'estimation de la posture du robot fournie par l'INS afin de pouvoir corriger les erreurs de dérive du gyroscope. Ainsi, la fonction de mesure $h(\delta\hat{\mathbf{X}}_k^-)$ est modifiée afin d'inclure la prédiction des angles, comme le montre l'équation (3.25).

$$\mathbf{h}(\delta\hat{\mathbf{X}}_k^-) = \begin{bmatrix} \|\mathbf{S}_1 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\| \\ \vdots \\ \|\mathbf{S}_4 - (\delta\hat{\mathbf{P}}_k^- + \hat{\mathbf{P}}_{INS,k})\| \\ \boldsymbol{\Theta}_{INS}(k) \end{bmatrix} \quad (3.25)$$

L'algorithme 3 présente la séquence d'exécution de l'algorithme de l'intégration UWB dans le filtre de Kalman Étendu.

```

initialization;
while data IMU available do
     $\tilde{\mathbf{u}}_k \leftarrow \text{read}(\text{accelerometer}, \text{gyroscope});$ 
     $\text{step} \leftarrow f_{ins\_orientation}(\tilde{\mathbf{u}}_k, \hat{\mathbf{X}}_k(7:9));$ 
     $\hat{\mathbf{X}}_k(7:9) \leftarrow \hat{\mathbf{X}}_k(7:9) + \text{step};$ 
    Compute  $E_k$ ;
    if data UWB available then
         $\delta\mathbf{z}_k \leftarrow [\text{dist}_{barycentre} \boldsymbol{\Theta}_{UWB}(k)];$ 
         $\mathbf{K}_k \leftarrow \boldsymbol{\Sigma}_k^- \mathbf{H}_k^T [\mathbf{H}_k \boldsymbol{\Sigma}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1};$ 
         $\delta\hat{\mathbf{X}}_k \leftarrow \mathbf{K}_k (\delta\mathbf{z}_k - \mathbf{h}(\delta\hat{\mathbf{X}}_k^-));$ 
         $\boldsymbol{\Sigma}_k \leftarrow [\mathbf{I}_{12 \times 12} - \mathbf{K}_k \mathbf{H}_k] \boldsymbol{\Sigma}_k^-;$ 
         $\hat{\mathbf{X}}_k(1:9) \leftarrow \hat{\mathbf{X}}_k(1:9) + \delta\hat{\mathbf{X}}_k(1:9);$ 
         $\delta\hat{\mathbf{X}}_k(1:9) \leftarrow 0;$ 
    end
     $\hat{\mathbf{X}}_{k+1} \leftarrow f_{ins}(\hat{\mathbf{X}}_k, \tilde{\mathbf{u}}_k);$ 
     $\boldsymbol{\Sigma}_{k+1} \leftarrow \mathbf{E}_k \boldsymbol{\Sigma}_k \mathbf{E}_k^T + \mathbf{Q}_k;$ 
end

```

Algorithm 3: Algorithme de l'intégration UWB pour l'estimation de l'orientation : La fonction $f_{ins_orientation}$ estime l'orientation du mobile en utilisant les données provenant de l'accéléromètre et le gyroscope et retourne la variation de l'angle par rapport à l'ancienne estimation. Une fois que les mesures de distance des trois *Tag* sont disponible, nous calculons l'orientation du robot $\boldsymbol{\Theta}_{UWB}$ en utilisant l'équation (3.23). Ensuite, l'erreur dans le vecteur d'état est estimée en calculant la différence entre le vecteur $\delta\mathbf{z}_k$ qui contient les mesures de distance qui sépare le barycentre du robot des différents balises avec la prédiction des mesures fournies par l'INS (3.25).

3.2.5 Performances de la localisation multi-capteurs

Dans le but d'estimer la précision du système de localisation multi-capteurs avec l'estimation de l'orientation *UWB*, nous avons commencé par la modélisation d'une trajectoire carrée sur Matlab comme le montre la Figure 3.12. À partir de cette trajectoire, nous calculons les positions réelles des modules *UWB* montés sur le robot formant un triangle équilatéral unitaire. Une fois que nous obtenons les trajectoires réelles parcourues par les modules *UWB*, nous appliquons sur ces dernières un bruit blanc gaussien avec une déviation standard de 10 cm.

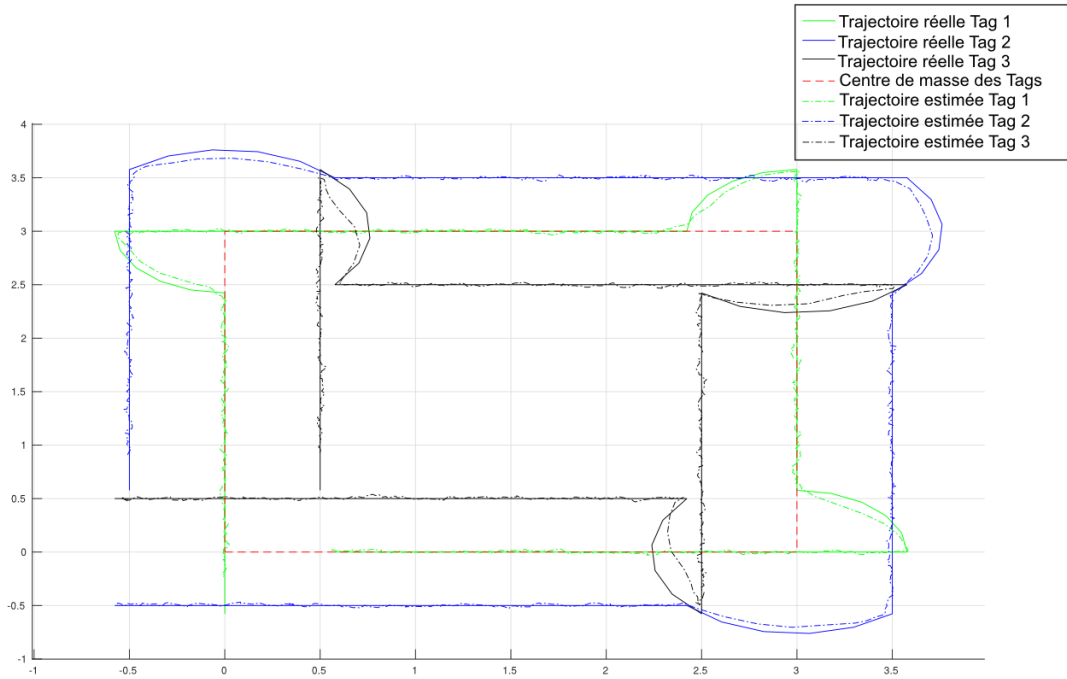


Figure 3.12 Résultats de la simulation du positionnement et de l'orientation avec les modules *UWB*.

Ensuite, dans la simulation nous appliquons sur ces mesures de distance notre algorithme de localisation pour estimer les positions des *Tags* (récepteurs) sur le robot. Les Figures 3.12 et 3.13 présentent les résultats obtenus en comparaison avec les valeurs réelles. Avec ce système de localisation, nous avons obtenu une déviation standard de 4 cm sur l'estimation de la position du mobile.

Finalement, nous utilisons les résultats obtenus pour estimer l'orientation du mobile en calculant la matrice de rotation entre le repère de navigation et le repère du robot. La Figure 3.14 présente une comparaison entre l'orientation réelle du robot et celle obtenue avec l'algorithme

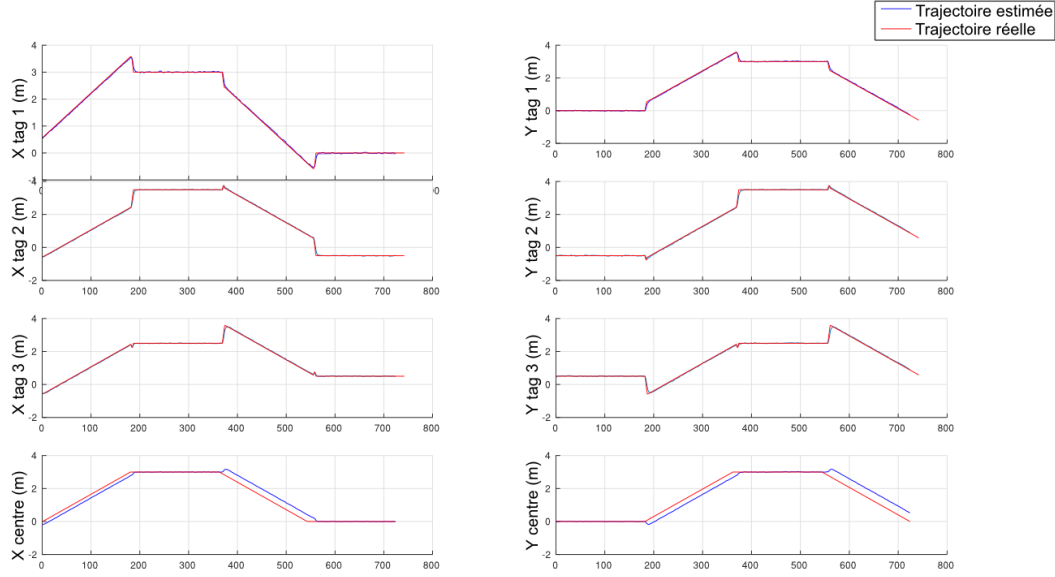


Figure 3.13 Résultats de la simulation du positionnement avec les modules UWB.

multi-capteur. La précision de calcul obtenue est de l'ordre de 2° . Cette estimation est utilisée par notre algorithme comme vecteur d'observation pour corriger les erreurs d'estimation de l'*INS*.

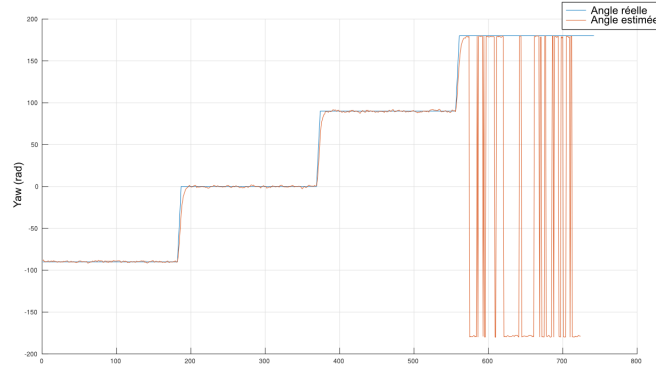


Figure 3.14 Résultats de la simulation de l'orientation avec le système *UWB*.

3.3 Conclusion

Dans ce chapitre, nous avons discuté dans un premier temps de la conception d'un système de localisation basé sur les mesures de distance UWB présentées au chapitre 2. Nous avons

démontré qu’avec des techniques classiques de localisation par “ranging” nous avons pu obtenir une estimation de position avec un faible taux d’erreur. L’avantage de l’utilisation de ces algorithmes est qu’ils peuvent être facilement implémentés sur des plates-formes embarquées avec une capacité de calcul limitée. Dans la deuxième partie, nous avons détaillé l’algorithme de fusion des données provenant du système de navigation inertielle et du système de mesure de distances UWB. Les résultats ont montré que le système de localisation développé conduit à une estimation précise de la posture du robot avec un taux de rafraîchissement suffisant pour effectuer des missions de navigation, en particulier dans les environnements intérieurs. Nous avons démontré également qu’une estimation de l’attitude d’un robot au sol est possible en l’absence de magnétomètre avec un niveau de précision de l’ordre de quelques degrés en utilisant trois modules UWB.

CHAPITRE 4 APPLICATIONS

Dans ce chapitre, nous validons finalement le système de positionnement *UWB* et les algorithmes développés en présentant les résultats obtenus expérimentalement. Nous avons conçu deux expériences sur deux types robot afin d'évaluer la performance de la localisation. Nous décrivons dans un premier temps l'intégration du système sur un nano-quadricoptère. Nous présentons ensuite l'intégration du système sur un véhicule autonome à roues afin d'estimer sa position et son orientation.

4.1 Intégration du système de localisation *UWB* sur un nano quadricoptère

Dans cette section, nous décrivons une expérience visant à évaluer le système proposé et son aptitude à contrôler des quadricoptères dans un environnement intérieur. Nous avons choisi comme plate-forme d'évaluation le drone Crazyflie 2.0¹, présenté par la Figure 4.1. Ce drone est l'une des plus petites plates-formes volantes disponibles sur le marché.



Figure 4.1 Intégration du module *DWM1000* sur le drone

Plusieurs caractéristiques font du Crazyflie 2.0 un candidat idéal pour évaluer la robustesse et la capacité de notre système à être utilisé avec une telle plate-forme complexe et dynamique. En effet, la petite taille avec un poids de 27 g, la charge utile limitée qui ne dépasse pas les

1. <https://www.bitcraze.io/crazyflie-2/>

15 g et le temps de vol d'environ 7 minutes présentent de nombreux défis en termes d'ajout de capteurs supplémentaires et de contrôle de vol.

4.1.1 Description de l'architecture logicielle et matérielle du quadricoptère

Le Crazyflie 2.0 est équipé d'un microcontrôleur du type STM32F405². Il est utilisé pour la gestion du système de vol (lecture des données des différents capteurs, estimation d'orientation, contrôleur de vol). Le drone dispose d'un autre microcontrôleur du type nRF51822³, utilisé pour la gestion de la distribution de puissance des quatre moteurs et de la communication radio avec le module Crazyradio. Le Crazyflie 2.0 est équipé également de plusieurs capteurs, un *IMU* qui contient un gyroscope et un accéléromètre trois axes, un magnétomètre et un baromètre. Cependant, le magnétomètre n'est pas intégré dans l'algorithme d'estimation d'orientation ce qui cause une divergence de l'angle de lacet au fil du temps. Cette plate-forme prévoit également un moyen facile pour l'intégration des nouveaux capteurs. La Figure 4.1 présente l'intégration du module *DWM1000* sur le drone avec le circuit imprimé qui a été développé à cette fin.

Un grand avantage de cette plate-forme est que toute la partie logicielle est à code source ouvert, ce qui facilite l'intégration d'autres capteurs. Cette architecture est basée sur le système d'exploitation temps réel FreeRTOS⁴. Elle exécute l'application principale composée de plusieurs services appelés "tasks", tels que la réception des commandes envoyées au quadricoptère, et le contrôleur PID. L'autre avantage est que les firmwares des deux microcontrôleurs nRF51822 et STM32F405 sont modifiables. Cependant, seul le firmware du microcontrôleur STM32F405 a été modifié pour la réalisation de cette expérience.

4.1.2 Réalisation

Dans une telle plate-forme volante, une haute fréquence de mise à jour du système de positionnement est nécessaire pour mieux contrôler le drone. D'autre part, en raison de la limitation de la taille des paquets envoyés à partir du Crazyflie au station de sol, avec une fréquence maximale de 100 Hz, il serait difficile d'atteindre la fréquence escomptée du système de positionnement avec un algorithme non embarquées sur le drone. En conséquence, nous avons opté de garder l'algorithme de mesure de distance et de localisation à bord du drone. Ainsi, nous pouvons réduire la charge de communication entre le drone et la station de contrôle.

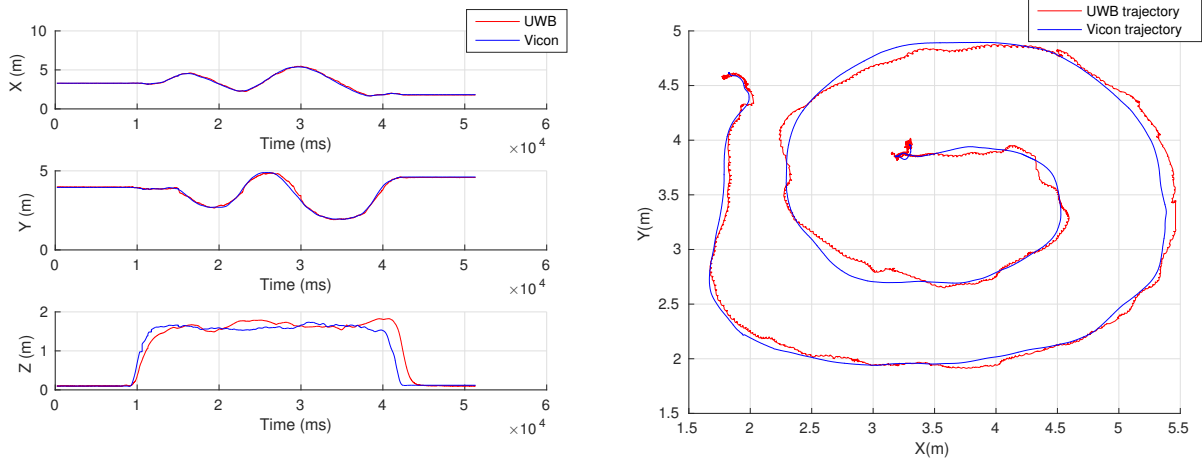
2. <http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN1035/PF252144>

3. <https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822>

4. <http://www.freertos.org/>

De plus, le drone peut activer le mode de vol stationnaire en cas de perte ou de perturbation de la communication. La capacité de calcul limitée du drone nous a aussi empêché d'utiliser l'algorithme de localisation multi-capteurs détaillé dans le chapitre 3. Ainsi, nous avons implémenté l'algorithme de filtrage de Kalman utilisant uniquement les mesures de distance *UWB* comme source d'information pour la localisation, comme détaillé à la Section 3.2.2. En effet, cet algorithme a montré de meilleurs résultats par rapport à l'algorithme de localisation multi-capteurs en termes de vitesse de convergence et de temps de calcul.

Afin d'évaluer la précision du système de positionnement embarqué sur le drone, nous utilisons le système de capture de mouvement optique VICON qui fournit une précision de 1 mm ou mieux. Ce système est basé principalement sur des caméras infrarouges placées à des positions fixes dans la zone d'exploration. Des marqueurs réfléchissants sont montés sur le drone afin de reconstruire la trajectoire effectuée en trois dimensions. Les Figures 4.2a et 4.2b présentent une comparaison des résultats de localisation obtenus par la système de positionnement *UWB* et le système VICON. Nous pouvons remarquer que le système *UWB* présente de meilleurs résultats sur les axes horizontaux par rapport à l'estimation d'altitude. Cela est dû aux erreurs sur les mesures de distance *UWB* provenant des sources d'erreur mentionnées au chapitre 2 et aussi à l'amplification de l'erreur à cause de la géométrie des balises utilisée dans cette expérience qui n'est pas bien adaptée pour donner une bonne estimation d'altitude. Nous utilisons quatre balises formant un rectangle, placées à une hauteur de 2.5 m. Les résultats de simulation ont montré qu'il est difficile de donner une bonne estimation d'attitude avec des balises placées à une telle hauteur, trop faible. Une solution pour remédier à ce problème est d'ajouter d'autres balises à différentes hauteurs. Manifestement, il s'agit de trouver un compromis entre la vitesse de convergence de l'algorithme de localisation et la précision de positionnement. Un autre problème qui affecte considérablement les résultats obtenus est la présence des obstacles dans la zone de travail. Au cours de déplacement du drone, il peut passer par des zones d'ombre ou moins de quatre balises sont disponibles pour la résolution du problème. De plus, la perte de ligne de vue direct entre l'antenne du drone et les différentes balises augmente l'erreur du positionnement [Denis et al. (2003)]. Le drone est équipé d'un module *UWB* de type *DWM1000* qui utilise une antenne unidirectionnelle pour la communication. Ainsi, un biais qui varie en fonction de l'angle d'orientation entre l'émetteur et le récepteur s'ajoute sur les mesures de distance.

(a) Comparaison du positionnement *UWB* et *VICON*

(b) Trajectoire effectué par le drone

Figure 4.2 Résultats des tests du système de positionnement UWB avec le Crazyflie 2.0

4.2 Navigation 2D en l'absence de magnétomètre

Dans cette section nous décrivons une autre expérience réalisée dans le but d'évaluer l'algorithme de localisation multi-capteurs. Nous définissons dans une première partie l'architecture matérielle et logicielle utilisée. Ensuite, nous présentons une évaluation des performances de notre système de localisation et nous discutons des résultats obtenus.

4.2.1 Description de l'architecture du système

La plate-forme de test utilisée pour cette expérience est un robot au sol. Une telle plate-forme présente une dynamique moins complexe et nécessite un taux de rafraîchissement moins élevé par rapport à un quadricoptère. Le robot Husky⁵ est utilisé pour les fins de cette expérience. Ce système robotique fournit une interface série permettant d'intégrer plusieurs types de capteurs à bord du robot. De plus, il permet le prototypage rapide des systèmes autonomes grâce au système open-source ROS (Robot Operating System⁶) qui facilite la tâche de développement logicielle.

ROS est méta-système d'exploitation qui fonctionne sur des systèmes d'exploitation UNIX conçus pour le développement logiciel des applications robotiques. Il contient une riche bibliothèque qui offre une abstraction matérielle de plusieurs types de capteurs ainsi qu'une interface de contrôle des périphériques de bas niveau. L'atout major de l'utilisation de cette

5. <http://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot/>

6. <http://www.ros.org/>

plate-forme est l'architecture souple de communication inter-processus et inter-machine ou plusieurs processus peuvent être exécutés sur des machines distinctes et qui peuvent communiquer de manière transparente pour l'utilisateur.



Figure 4.3 Configuration matérielle du système de localisation multi-capteurs

4.2.2 Réalisation

Trois *Tag* ont été utilisés pour la réalisation de cette expérience afin d'estimer l'orientation du robot à partir des mesures de distance. Ces modules sont placés de façon à former un triangle équilatéral avec des côtés de longueur de 1 m. Les nœuds *UWB* sont reliés au robot via l'interface de communication série.

La Figure 4.4 présente un schéma simplifié de l'architecture des nœuds ROS implantés sur le robot. Une séparation est faite entre la couche de communication qui prend en charge

la récupération des données des différents capteurs et la couche application qui contient l'implémentation de l'algorithme de fusion des données des capteurs.

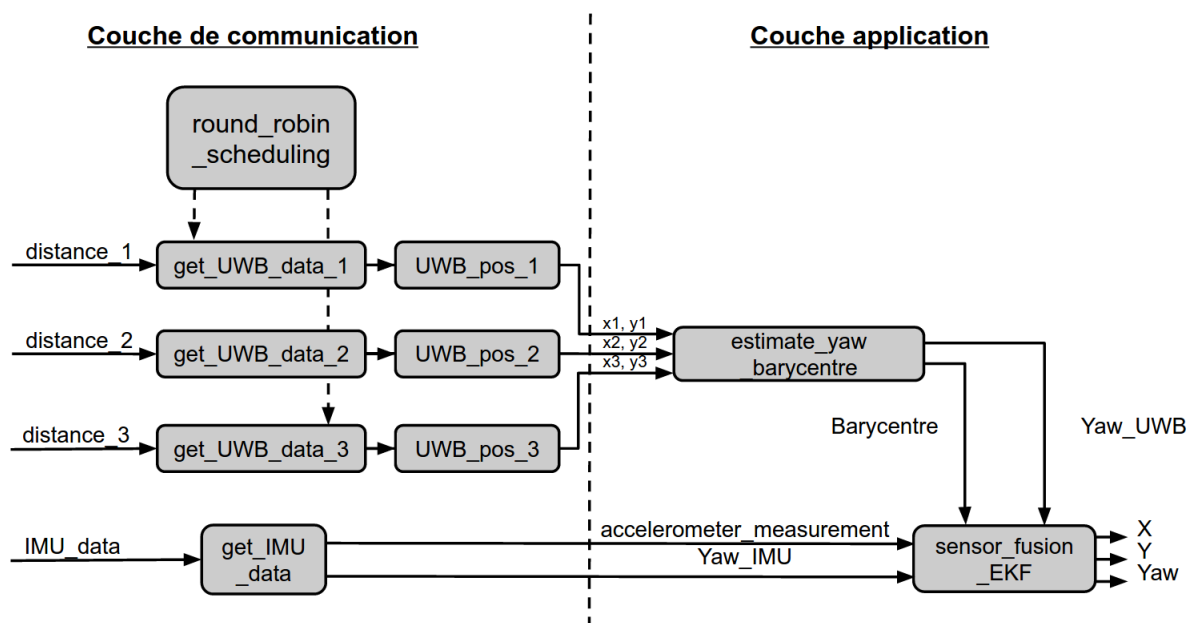


Figure 4.4 Architecture des nœuds ROS

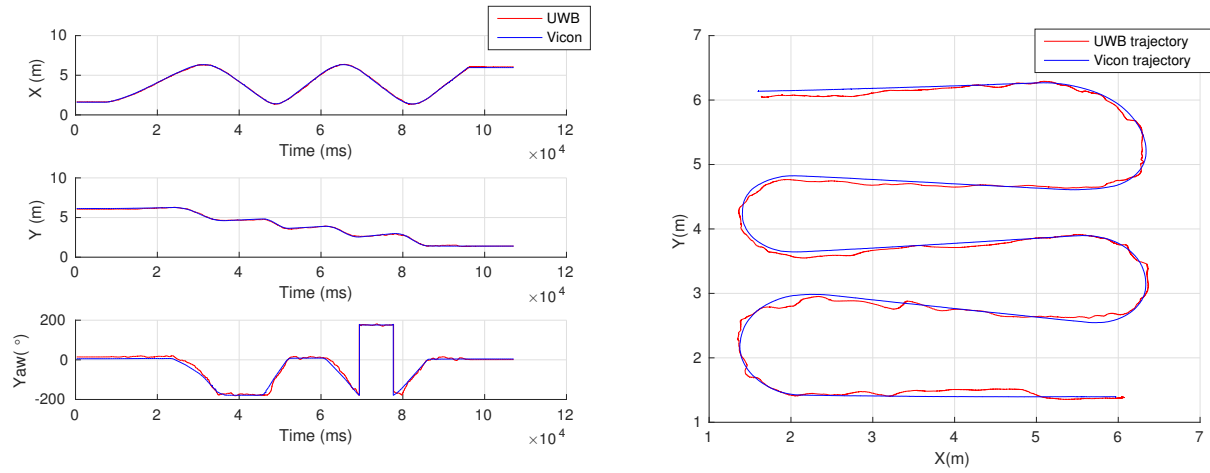
Dans la couche de communication, le nœud ROS *round_robin_scheduling* se charge de la planification de la récupération des données des modules *UWB*, étant donné que les Tags montés sur le robot ne peuvent pas lancer un échange de messages avec les balises tous en même temps. L'idée consiste à matérialiser le droit de la communication avec les balises par la possession d'un jeton (token) unique que les Tags peuvent se transmettre. Ce jeton circule entre les différents Tags périodiquement. Celui qui a le jeton est le seul qui peut lancer un échange de messages avec les stations de base. L'estimation de la position se fait principalement sur deux nœuds ROS :

- *get_UWB_data_i* qui lance la connexion série avec le Tag afin de récupérer les mesures de distances qui le séparent des balises et *UWB_pos_i* qui résout le problème de la localisation en combinant les mesures de distances à l'aide d'un filtre de Kalman.
- Le nœud ROS *get_IMU_data* se charge de la récupération des données de l'accéléromètre et l'orientation estimée par l'*IMU* pour les publier à la couche de haut niveau.

La couche application contient l'implémentation des algorithmes de positionnement des Tags et de fusion de données. Le nœud ROS *estimate_yaw_barycentre* se met à l'écoute des mises à jour des positions des Tags. Une fois qu'il reçoit une estimation de position des trois Tags,

il lance le processus de l'estimation du barycentre du robot et de l'angle de l'orientation. Ces estimations sont ensuite publiées au nœud ROS *sensor_fusion_EKF* pour faire la fusion des données de l'IMU et des modules UWB en utilisant un filtre de Kalman Étendu, détaillé au chapitre 3.

La Figure 4.5 présente les résultats d'expérimentation du système de fusion de données UWB/INS intégrant l'estimation de l'orientation UWB. Ces résultats ont été comparés avec le système VICON afin d'évaluer la précision des estimations. Nous pouvons remarquer que cette algorithme présente moins d'erreur par rapport à l'algorithme de filtrage de Kalman basé uniquement sur les mesures de distance.



(a) Comparaison du positionnement *UWB* et *VICON*

(b) Trajectoire effectué par le robot

Figure 4.5 Résultats des tests du système de positionnement UWB avec le Husky

CHAPITRE 5 CONCLUSION

Dans ce mémoire, nous avons présenté nos contributions au domaine de la recherche en robotique mobile avec le développement d'un système de mesure de distance entre des robots équipés de modules UWB. Nous avons également démontré l'intégration de ce système pour l'estimation de la posture (position et orientation) avec deux types de plateformes robotiques : robot au sol et drone.

Dans ce chapitre, nous résumons les apports de ce travail en évaluant la performance obtenue pour notre système. Nous concluons en présentant les limitations et les futures perspectives du système proposé.

5.1 Synthèse des travaux

Le système proposé permet d'effectuer des mesures de distance entre deux équipements radio avec une précision de l'ordre de quelques centimètres. Un tel système peut servir à une variété d'applications, notamment dans les algorithmes multi-robots, tels que l'algorithme de rendez-vous. Cet algorithme consiste à rassembler plusieurs robots à un point d'arrivée donné, en partant de positions initialement dispersées dans la zone de travail. Les mesures de distance précises nous ont permis de concevoir un système de positionnement permettant de localiser un robot mobile dans une zone d'exploration avec une haute précision, de l'ordre du centimètre. Par ailleurs, la fusion des mesures de distance UWB avec un système de navigation inertielle permet d'estimer l'attitude du robot et d'améliorer considérablement le taux de rafraîchissement du système global. La mesure du cap avec une précision de l'ordre de quelques degrés a aussi été démontrée par l'installation de plusieurs tags sur le robot. Ces fonctionnalités ouvrent la voie notamment à la réalisation des missions de navigation autonome.

Durant mes travaux, plusieurs méthodes de localisation ont été développées dans le but de satisfaire les objectifs du projet. Nous avons implémenté quatre algorithmes de localisation. Deux d'entre eux utilisent seulement les mesures de distance entre les modules UWB pour en déduire la position relative du mobile. L'avantage de l'utilisation de ces algorithmes consiste en sa facilité de déploiement sur des plates-formes embarquées avec une capacité de calcul limitée. Les deux autres algorithmes fusionnent les mesures de distance UWB avec les données d'un accéléromètre et d'un gyroscope provenant d'une centrale inertielle. Une telle méthode permet de profiter des avantages de chaque capteur individuellement afin d'avoir une meilleure

maîtrise sur la navigation. Les résultats de l'évaluation en simulation et en expérimentation de notre système de localisation ont montré que les algorithmes se basant sur plusieurs sources d'informations sont beaucoup plus performants en terme de précision de calcul et de fréquence d'exécution.

5.2 Limitations et futures perspectives de la solution proposée

La solution adoptée dans notre système de mesure de distance se base un module UWB équipé d'une antenne unidirectionnelle. Le non-alignement des antennes de réception et d'émission peut induire des erreurs significatives pendant le calcul de la distance. Nous proposons d'intégrer à la puce *DW1000* une antenne omni-directionnelle, ce qui va étaler le diagramme de rayonnement de l'antenne à 360 degrés et augmenter la couverture du module. Ainsi, nous pourrions réduire l'erreur sur les mesures de distance et par conséquent améliorer la précision du positionnement du robot. Dans le chapitre 2, nous avons présenté une caractérisation de l'impact de la directionnalité des antennes sur les mesures de distance. Nous proposons d'intégrer cette caractérisation à l'algorithme de filtrage de ces mesures. Cette fonctionnalité peut donc compenser l'erreur notamment dans des conditions de visibilité directe entre les modules.

Le système de localisation développé exige une connaissance à l'avance de nombre et de l'emplacement des stations de base dans le déplacement. Afin d'automatiser cette tâche, nous proposons d'ajouter un module de découverte du voisinage. Ce module aura comme rôle l'exploration de nouvelles zones de déplacement en partant d'une configuration initiale. Dès que le robot détecte la perte d'une station de base, il lance le processus de découverte de voisinage en diffusant son identifiant dans le réseau et se met en attente des réponses. À la réception d'une tentative de connexion, les stations de base doivent renvoyer à l'émetteur leurs adresses ainsi que leur emplacement. Enfin, l'émetteur met à jour sa base de données avec les coordonnées des balises et relance l'algorithme de localisation.

Un autre champ d'amélioration est l'auto-positionnement des stations de base UWB dispersées dans la zone de travail en utilisant les distances que celles-ci peuvent mesurer entre elles. En effet, actuellement l'emplacement de ces balises doit être fait avec précaution à une précision centimétrique pour ne pas affecter l'estimation de la position des nœuds mobiles. L'auto-positionnement permettra également de mettre en place rapidement et de localiser un nombre relativement important de stations de base UWB, en ne localisant manuellement que quelques-unes d'entre elles.

Le système de localisation intégrant l'estimation de l'orientation UWB a été testé seulement

en deux dimensions tandis que nos algorithmes de fusion de données ont été développés avec une modélisation de déplacement en trois dimensions. Il serait intéressant de tester ce système dans un environnement de déplacement en trois dimensions.

RÉFÉRENCES

- P. Benjamin, “Development of an automatic imu calibration system”, *ETH Master Thesis*, 2013.
- A. Bensky, *Wireless Positioning : Technologies and Applications*. Artech House, 2007.
- J. Borràs Sillero, “Sensor fusion methods for indoor navigation using uwb radio aided ins/dr”, 2012.
- A. Bridgewater, “Analysis of second and third order steady-state tracking filters”, dans *AGARD Conference Proc*, no. 252, 1978, pp. 9–1.
- J. J. Caffery Jr, *Wireless location in CDMA cellular radio systems*. Springer Science & Business Media, 2006, vol. 535.
- DecaWave, “Dwm1000 datasheet”, 2014.
- , “Sources of error in DW1000 based two-way ranging (TWR) schemes”, 2014.
- , “Antenna delay calibration of dw1000-based products and systems”, 2014.
- B. Denis, J. Keignart, et N. Daniele, “Impact of nlos propagation upon ranging precision in uwb systems”, dans *Ultra Wideband Systems and Technologies, 2003 IEEE Conference on*. IEEE, 2003, pp. 379–383.
- J. Farrell, *Aided navigation : GPS with high rate sensors*. McGraw-Hill, Inc., 2008.
- S. Gezici, H. Kobayashi, et H. V. Poor, “A new approach to mobile position tracking”, dans *Proc. IEEE Sarnoff Symp. Advances in Wired and Wireless Communications*, 2003, pp. 204–207.
- S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, et Z. Sahinoglu, “Localization via ultra-wideband radios : a look at positioning aspects for future sensor networks”, *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70–84, 2005.
- S. Gratton, A. S. Lawless, et N. K. Nichols, “Approximate Gauss-Newton methods for nonlinear least squares problems”, *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 106–132, 2007.

- P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*, 2e éd. Artech House, 2013.
- B. Kempke, P. Pannuto, et P. Dutta, “Polypoint : Guiding indoor quadrotors with ultra-wideband localization”, dans *HotWireless’15*, 2015.
- M. P. Layer, “Part 15.4 : Low-Rate Wireless Personal Area Networks (LR-WPANs)”, 2013.
- A. Ledergerber, M. Hamer, et R. D’Andrea, “A robot self-localization system using one-way ultra-wideband communication”, dans *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seattle, Washington, September 2015, pp. 3131–3137.
- H. Liu, H. Darabi, P. Banerjee, et J. Liu, “Survey of wireless indoor positioning techniques and systems”, *IEEE Transactions on Systems, Man, and Cybernetics - Part C : Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, November 2007.
- , “Survey of wireless indoor positioning techniques and systems”, *Systems, Man, and Cybernetics, Part C : Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, 2007.
- S. O. H. Madgwick, A. J. L. Harrison, et R. Vaidyanathan, “Estimation of IMU and MARG orientation using a gradient descent algorithm”, pp. 1–7, 2011.
- M. McGuire, K. N. Plataniotis, et A. N. Venetsanopoulos, “Location of mobile terminals using time measurements and survey points”, *Vehicular Technology, IEEE Transactions on*, vol. 52, no. 4, pp. 999–1011, 2003.
- M. W. Mueller, M. Hamer, et R. D’Andrea, “Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadcopter state estimation”, dans *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hamburg, Germany, May 2015, pp. 1730–1736.
- C. Nerguizian, C. Despins, et S. Affès, “Geolocation in mines with an impulse response fingerprinting technique and neural networks”, *Wireless Communications, IEEE Transactions on*, vol. 5, no. 3, pp. 603–611, 2006.
- D. Niculescu et B. Nath, “Ad hoc positioning system (APS) using AOA”, dans *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3. Ieee, 2003, pp. 1734–1743.

J.-O. Nilsson, A. De Angelis, I. Skog, P. Carbone, et P. Händel, “Signal processing issues in indoor positioning by ultra wide band radio aided inertial navigation”, pp. 2161–2165, 2009.

A. Shareef et Y. Zhu, “Localization using extended Kalman filters in wireless sensor networks”, 2009.

Y. Shimizu et Y. Sanada, “Accuracy of relative distance measurement with ultra wide-band system”, *Electronics and Communications in Japan(Part III Fundamental Electronic Science)*, vol. 87, no. 12, pp. 26–36, 2004.

B. Tippetts, D. J. Lee, K. Lillywhite, et J. Archibald, “Review of stereo vision algorithms and their suitability for resource-limited systems”, *Journal of Real-Time Image Processing*, vol. 11, no. 1, pp. 5–25, 2016.

C. Troiani, A. Martinelli, C. Laugier, et D. Scaramuzza, “Low computational-complexity algorithms for vision-aided inertial navigation of micro aerial vehicles”, dans *Robotics and Autonomous Systems Journal*, vol. 69, 2015, pp. 80–97.

D. Vasisht, S. Kumar, et D. Katabi, “Sub-nanosecond time of flight on commercial wi-fi cards”, Massachusetts Institute of Technology, Rapp. tech., May 2015, available at <http://arxiv.org/pdf/1505.03446.pdf>.

H. Wymeersch, S. Maranò, W. M. Gifford, et M. Z. Win, “A machine learning approach to ranging error mitigation for UWB localization”, *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1719–1728, June 2012.

R. Yarlagadda, I. Ali, N. Al-Dhahir, et J. Hershey, “GPS GDOP metric”, *IEE Proceedings : Radar, Sonar and Navigation*, vol. 147, no. 5, pp. 259–264, 2000.

ANNEXE A Architecture logicielle et matérielle du système UWB

A.1 Architecture logicielle

La Figure A1 présente la structure logicielle de notre application qui est composée de plusieurs couches. Une couche de bas niveau contient la configuration du périphérique *SPI* et la configuration de la ligne d'interruption de la carte *STM32F3*. Cette couche fait une abstraction de la configuration matérielle qui varie d'un microcontrôleur à un autre, ce qui facilite la tâche du changement de carte de développement selon les besoins. Elle offre deux fonctions génériques *Lecture_SPI()* et *Écriture_SPI()* qui vont être utilisées par les couches de plus haut niveau afin d'effectuer des lectures et des écritures du module ultra-wide band. Il y a également une autre couche qui contient le pilote du module *DW1000*. Cette couche nous a été livrée par la compagnie DecaWave afin de faciliter la configuration du module, et la license pour ce driver appartient à *DecaWave*.

La couche de plus haut niveau est la couche application; elle contient des structures des données qui nous permettent de changer facilement les modes de fonctionnement du module moyennant le bloc *Initialisation des modes*, comme par exemple le changement de la vitesse de transmission des trames, la longueur des trames, etc. Elle contient aussi l'implémentation de la machine à états qui exécute l'algorithme de mesure de distance, ainsi que des fonctions de rappel provenant du bloc *Traitement de l'interruption* qui envoie à son tour des notifications à la couche application suite à une réception ou une émission d'une trame. Nous avons donc une forte dépendance entre la machine à états et le traitement de l'interruption, ce qui permet aux fonctions rappel de mettre à jour l'état de la machine en fonction des trames reçues où envoyées.

A.1.1 Machines à états

Machine à états du *Tag*

Le déroulement de notre algorithme de mesure de distance se fait principalement dans deux processus, la machine à états et le traitement de l'interruption. Dans la machine à états A2, nous commençons par lancer l'initialisation qui consiste à faire quelques configurations qui ne sont pas communes entre le *Tag* et l'*Anchor* comme par exemple l'affectation d'une adresse au module. Ensuite, nous passons au prochain état dans lequel nous envoyons la première trame de l'échange qui contient un message de type *Poll*. Nous déclenchons également dans le même état un timer qui lance un nouvel échange après l'expiration du temps alloué si

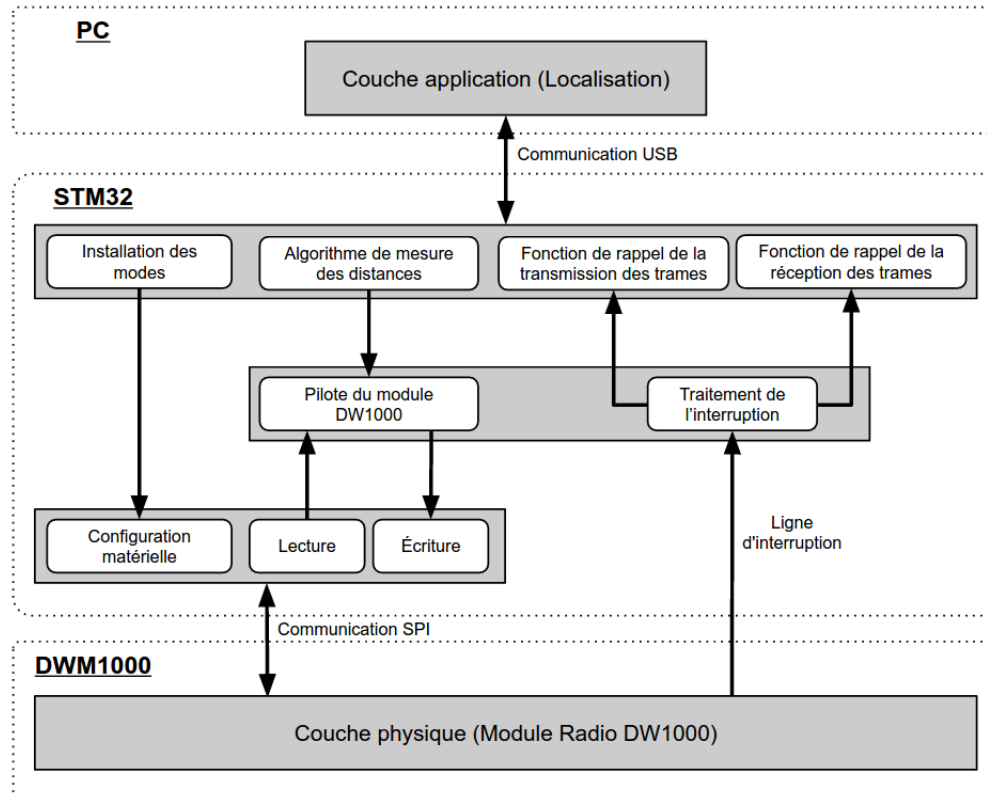
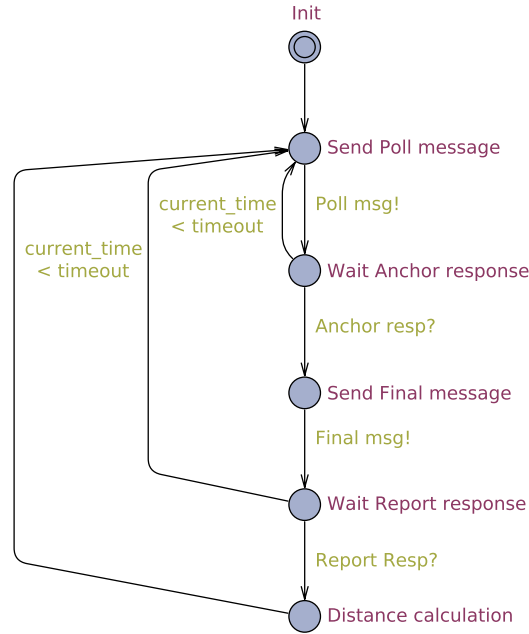


Figure A1 Description de la structure logicielle

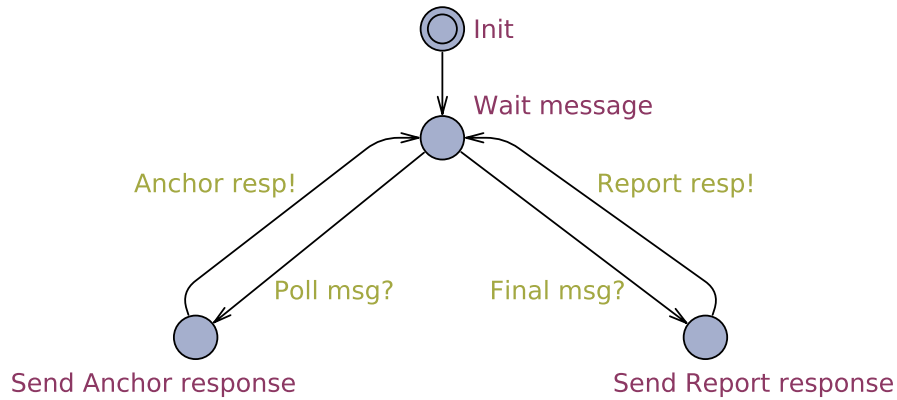
jamais il y a eu des problèmes durant l'échange courant. Une fois que nous avons reçu la confirmation de l'envoi du message, nous passons à un état d'attente d'une réponse qui doit être une trame de type *Anchor Response* en réponse à la trame du type *Poll* envoyée par le *Tag*. À la réception de cette réponse, le module *DW1000* active la ligne d'interruption afin de notifier le microcontrôleur de cet événement pour qu'il lance le processus du traitement de l'interruption dans lequel nous sauvegardons le timestamp de la trame et nous passons à l'état *Send Final message*. Dans cet état, nous envoyons la dernière trame de l'échange et nous passons de nouveau à un état d'attente d'une réponse qui doit être une trame de type *Report*. À la réception de cette trame qui contient tous les timestamps de l'*Anchor*, le *Tag* calcule la distance qui le sépare de l'*Anchor* et lance un nouvel échange.

Machine à états du *Anchor*

De la même façon que dans la machine à état du *Tag*, nous commençons par l'initialisation comme le montre la figure A3 et nous passons à un état d'attente de réception des trames provenant du *Tag*. Suite à la réception d'une trame, l'*Anchor* répond en fonction du type de

Figure A2 Machine à états du *Tag*.

message reçu. Il envoie un message du type *Anchor Response* s'il reçoit un message du type *Poll* et un message du type *Report* s'il reçoit le message *Final Response* de la part de *Tag*. De cette façon, nous pouvons garantir une indépendance entre le *Tag* et l'*Anchor* qui nous permet d'éviter de faire une synchronisation entre les deux stations. Nous avons opté pour cette solution car la synchronisation peut causer des problèmes d'interblocage.

Figure A3 Machine à états du *Anchor*.

A.1.2 Protocole de communication entre les modules UWB

Le format général d'une trame utilisée dans les modules *DW1000* suit la norme IEEE 802.15.4 (Layer (2013)), comme le montre le tableau A1. Le champ de contrôle de trame contient deux octets qui nous permettent de spécifier le mode de fonctionnement des modules, le mode standard qui utilise des trames de données avec 8 octets (64 bits) pour encoder les adresses de l'émetteur et du récepteur, et le mode rapide qui utilise seulement deux octets (16 bits) pour l'encodage des adresses.

Tableau A1 Format général d'une trame

2 octets	1 octet	2 octets	8 octets	8 octets	# octets	2 octets
Contrôle de trame (FC)	Numéro de séquence	PAN ID (0xCADE)	Adresse de destination	Adresse de l'émetteur	Champ de données	FCS

Le champ de données de la trame est géré par l'utilisateur en fonction de l'application. Il contient l'information qui circule entre l'émetteur et le récepteur. Dans notre application, nous utilisons quatre types de message afin de décoder les états de l'échange de messages dans notre algorithme de mesure de distance :

1. Message de type *Poll*

Le message de type *Poll* présenté par la table A2 est envoyé par le *Tag* pour initialiser l'échange de messages entre lui et l'*Anchor*. Il dispose d'un seul octet avec la valeur (0x21) qui permet au récepteur de répondre avec le bon message.

L'utilisation du champ de données optionnel varie en fonction de l'application, par exemple pour effectuer une mesure de distance, ce champ peut être vide. Nous pouvons également ajouter d'autres informations qui ne sont pas en rapport avec les mesures de distance, ce qui nous donne des futures possibilités d'expansion.

Tableau A2 Message de type *Poll*

Code du message	Champ de données (optionnel)
0x21	-

2. Message de type *Anchor Response*

Le message de type *Anchor Response* (table A3) est envoyé par l'*Anchor* suite à une réception d'un message *Poll*. Il contient un seul octet avec la valeur (0x10) et un champ de données optionnel.

Tableau A3 Message de type *Anchor Response*

Code du message	Champ de données (optionnel)
0x10	-

3. Message de type *Final Response*

Le message du type *Final Response* (table A4) est envoyé par le *Tag* à l'*Anchor* suite à une réception d'un message du type *Anchor Response*. Et comme son nom l'indique, c'est le dernier message de la part du *Tag* dans un échange.

Tableau A4 Message de type *Final Response*

Code du message	Champ de données (optionnel)
0x29	-

4. Message de type *Report*

Ce message (table A5) est envoyé par l'*Anchor* afin de reporter au *Tag* les temps de réception et d'envoi qui sont sauvegardés dans l'*Anchor*. Il contient le code du message (0x2A) suivi de 5 octets du temps de réception du message *Poll*, 5 octets du temps d'envoi du message *Anchor Response* et de 5 octets du temps de réception du message *Final*.

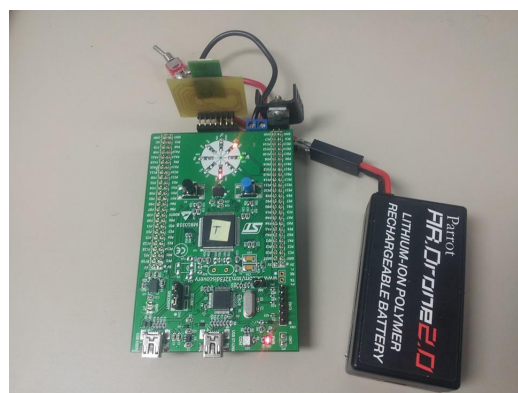
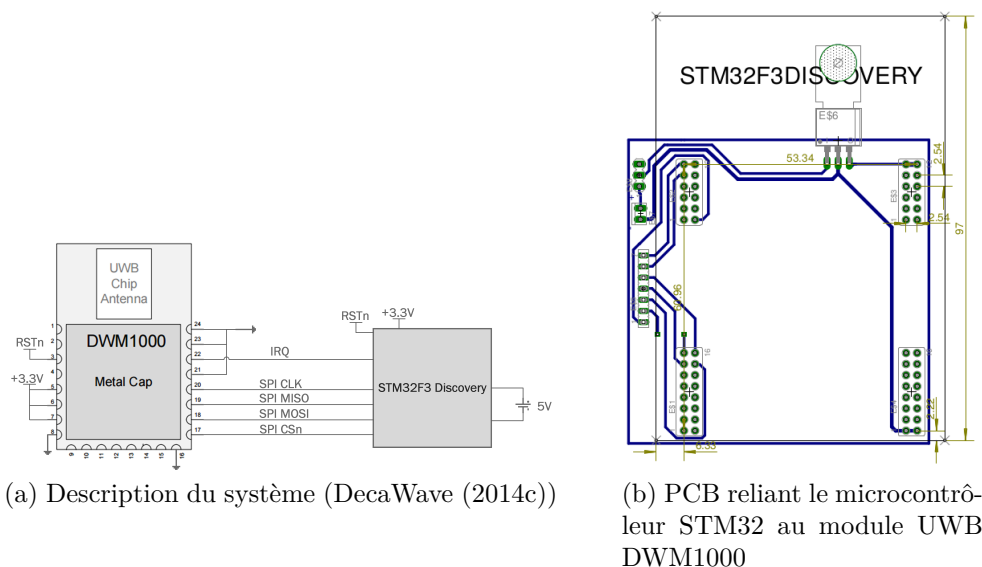
Tableau A5 Message de type *Report*

Code du message	Temps de réception du message <i>Poll</i>	Temps d'envoi du message <i>Anchor Response</i>	Temps de réception du message <i>Final</i>	Champ de données (optionnel)
0x2A	t'_1	t'_2	t'_5	-

A.2 Architecture matérielle des modules UWB

Les Figures A4a et A4b montrent les différents composants du système développé. Celui-ci est composé d'une carte avec microcontrôleur de type *STM32F3 Discovery*, qui abrite l'application de mesure de distances, et d'un module UWB *DWM1000* de *Decawave* intégrant la puce *DW1000* et une antenne. Une interface de communication *SPI* (*Serial Peripheral Interface*) relie ces deux composants et permet de configurer le module, envoyer et recevoir des trames. Il y a également une ligne d'interruption qui est activée quand une trame entre ou

quitte l'antenne. Cette interruption est utilisée dans l'application afin de récupérer le statut des trames envoyées ou reçues par le module. Quand le microcontrôleur envoie une trame au module à travers le *SPI*, ce dernier active la ligne d'interruption. Le microcontrôleur à son tour lance des lectures dans les registres qui contiennent le statut de l'émission de la trame afin de vérifier s'il n'y a pas eu une erreur d'encodage du contenu de la trame et de récupérer le timestamp (à quelle heure la trame a quitté l'antenne). Nous traitons aussi de la même façon les trames reçues. Quand la ligne d'interruption s'active à la réception d'une trame, le microcontrôleur prend en charge la récupération de son contenu.



(c) Image du prototype du système de localisation

Figure A4 Exemple d'une station du système de localisation